

POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Meccanica

STUDIO DELL'INTERAZIONE TRA UOMO E ROBOT COLLABORATIVO E APPLICAZIONE ALLO SMONTAGGIO DELLA BATTERIA DI UN VEICOLO ELETTRICO



Relatori

Prof. Stefano Mauro
Ing. Matteo Melchiorre
Ing. Laura Salamina

Candidato

Francesco Laterza

Dicembre 2023

Sommario

Questo lavoro si concentra sulla robotica collaborativa e sulle sue caratteristiche principali. Ci si è soffermati sull'applicazione della robotica collaborativa per lo smontaggio della batteria di un veicolo elettrico. Nel caso in questione è stata scelta una Chevrolet Bolt 2017. Per il lavoro è stato usato un robot antropomorfo, nello specifico di un robot FR3 prodotto dalla Franka Emika, montato su un Autonomous Mobile Robot, nello specifico di un TiaGo Base, prodotto dalla Pal Robotics.

Per verificare l'efficacia del processo, si è inizialmente effettuata un'analisi volta a individuare la sequenza di smontaggio ottimale della batteria. Si è partiti dal diagramma di smontaggio della batteria e, con un algoritmo apposito realizzato con il software Matlab, si è stabilito quali task affidare a ciascun operatore e l'ordine con cui essi vanno eseguiti. Noto il processo di smontaggio, ci si è interessati alle coppie di task che uomo e robot eseguono nello stesso arco temporale. Sono state create delle casistiche di cui fanno parte coppie di task con proprietà simili tra loro. Per ogni casistica ci si è concentrati soprattutto sul tipo di interazione tra uomo e robot e si sono scelte le casistiche in cui l'intervento della robotica collaborativa è più evidente.

A tale scopo è stato implementato un algoritmo di collision avoidance basato su velocità repulsive che impediscono ai due robot di scontrarsi con l'uomo o di urtare eventuali ostacoli. L'algoritmo è realizzato con il software Matlab, mentre i movimenti sono stati riprodotti in ambiente 3D con il software CoppeliaSim, dove è stato anche riprodotto l'ambiente di lavoro con gli elementi principali in scala 1:1.

I risultati mostrano come l'applicazione della robotica collaborativa permette di alleggerire il carico di lavoro dell'uomo senza ripercussioni rilevanti in termini di tempo richiesto e di risorse impiegate, oltre che come gli algoritmi di robotica collaborativa hanno ancora ampi margini di miglioramento.

Indice

Elenco delle figure	3
Elenco delle tabelle	6
1 Introduzione	7
1.1 Caratteristiche peculiari	8
1.2 Applicazioni	9
1.3 Franka Emika FR3	10
1.4 Robot Mobili	11
1.5 Tiago Base	12
1.6 Electric Vehicle Battery	13
2 Analisi smontaggio batteria	17
2.1 Introduzione al problema	17
2.2 Tecniche per l'ottimizzazione dello smontaggio	18
2.3 Metodo di smontaggio con algoritmo di ottimizzazione	28
2.4 Spiegazione dell'algoritmo di ottimizzazione	33
3 Analisi Interazione Uomo Robot	37
3.1 Modello del TiaGo Robot	39
3.2 Campi Potenziali	40
3.3 Applicazione della Collision Avoidance al TiaGo	45
3.4 Applicazione della Normativa al TiaGo	54
3.5 Modello del Robot Franka Emika FR3	56
3.6 Percorso dell'End Effector senza Collision Avoidance	58
3.7 Percorso dell'End Effector con applicazione della Collision Avoidance	60
3.8 Applicazione della normativa al robot FR3	62
4 Risultati	69
4.1 Casistica 1	71
4.2 Casistica 2	75
4.3 Casistica 3	78
5 Conclusioni	83

Elenco delle figure

1.1	Robot Franka Emika FR3	10
1.2	AMR TiaGo Base	12
1.3	Varianti del Tiago in base agli elementi montati sopra	13
2.1	Schema per il riutilizzo di una batteria di un veicolo elettrico	18
2.2	Schema della batteria di un audi Q5	19
2.3	Batteria di un Audi Q5, con i vari componenti numerati	20
2.4	Matrice di smontaggio prioritario	21
2.5	Stazione di lavoro con operatore e braccio meccanico fissato sul tavolo	21
2.6	Interno di una batteria Chevrolet Bolt 2017	22
2.7	Diagramma di smontaggio della batteria della Chevrolet Bolt 2017 (parte 1)	23
2.8	Diagramma di smontaggio della batteria della Chevrolet Bolt 2017 (parte 2)	24
2.9	Diagramma di smontaggio della batteria della Chevrolet Bolt 2017 (parte 3)	24
2.10	Criteri per la valutazione dell'assegnazione dei task ai due operatori (uomo e robot)	25
2.11	Valori NAI e TAAi riferiti ai componenti della batteria dell'Audi Q5	26
2.12	Valori NAI e TAAi riferiti ai componenti della batteria dell'Audi Q5 (grafico)	26
2.13	Valori NAI e TAAi riferiti ai componenti della batteria della Chevrolet Bolt 2017	27
2.14	Valori NAI e TAAi riferiti ai componenti della batteria della Chevrolet Bolt 2017 (grafico)	28
2.15	Posizione di 3 task generici nello ambiente di lavoro	29
2.16	Insieme dei task delegabili all'uomo e insieme dei task delegabili a entrambi gli operatori	33
2.17	Insieme dei task effettuabili subito e insieme dei task non ancora effettuabili	34
2.18	Combinazione degli insiemi rappresentati in 2.16 e 2.17	34
2.19	Plottaggio di 80 tempi di simulazione con il nuovo algoritmo di ottimizzazione	35
2.20	Rappresentazione schematica dei due operatori nell'ambiente di lavoro	36
3.1	Ambiente di simulazione in Coppelasim con tutti gli elementi	38
3.2	Telecamera Kinect	39
3.3	Schema per il calcolo del "Optimized skeleton"	39
3.4	Posizione e orientamento del TiaGo nel piano	40
3.5	Campo potenziale attrattivo U_{base}	41

3.6	Campo potenziale repulsivo del manichino	43
3.7	Campo potenziale repulsivo del tavolo	43
3.8	Campo potenziale complessivo	44
3.9	Esempio di situazione in cui $(v_{Tot})=0$	45
3.10	Campo potenziale complessivo	45
3.11	Configurazione generica dei punti A e B con ostacolo di forma circolare . .	46
3.12	Scelta percorso ottimale (caso 1)	47
3.13	Scelta percorso ottimale (caso 2)	47
3.14	Scelta percorso ottimale (caso 3)	48
3.15	Scelta percorso ottimale (caso 4)	48
3.16	Configurazione generica dei punti A e B con ostacolo di forma rettangolare	49
3.17	Configurazione in cui $v_{Tot} = v_{base} + v_{rep}$	49
3.18	Configurazione in cui $ v_{Tot} = v_{base} $ indipendentemente da v_{rep}	50
3.19	Caso in cui non è necessario l'intervento di collision avoidance	50
3.20	Schema del TiaGo con il versore imposto dai campi potenziali e il versore effettivo del robot	53
3.21	Manichino e cilindro per l'applicazione della normativa	55
3.22	Robot FR3 in ambiente Coppeliasim	56
3.23	Robot F53 con i parametri di Denavit-Hartenberg	57
3.24	Posa del sistema di riferimento associato al giunto q_7 , fissa rispetto al sistema di riferimento associato al giunto q_6	57
3.25	Matrici di posa del robot FR3 (parte 1)	58
3.26	Matrici di posa del robot FR3 (parte 2)	59
3.27	Robot FR3 approssimato da sfere	64
3.28	Manichino approssimato da sfere e cilindri nella parte superiore del corpo .	64
4.1	Diagramma di Gantt della sequenza di smontaggio 1, tempo ciclo di 3625 s	70
4.2	Diagramma di Gantt della sequenza di smontaggio 2, tempo ciclo di 3627 s	70
4.3	Posizione dei due operatori ad inizio simulazione	72
4.4	Posizione dei due operatori a simulazione conclusa	72
4.5	Percorsi del TiaGo e dell'end effector del FR3 nello spazio cartesiano . . .	73
4.6	Campi Potenziali repulsivi	73
4.7	Confronto tra le velocità $ v_{avanz} $ delle sfere solidali al robot FR3 e la curva v_{norm}	74
4.8	Confronto tra le velocità $ v_{avanz} $ del TiaGo e la curva v_{norm}	75
4.9	Posizione dei due operatori ad inizio simulazione	75
4.10	Posizione dei due operatori a simulazione conclusa	76
4.11	Percorsi del TiaGo e dell'end effector del FR3 nello spazio cartesiano . . .	76
4.12	Campi Potenziali repulsivi	77
4.13	Confronto tra le velocità $ v_{avanz} $ delle sfere solidali al robot FR3 e la curva v_{norm}	77
4.14	Posizione dei due operatori ad inizio simulazione	78
4.15	Posizione dei due operatori a simulazione conclusa	79
4.16	Percorsi del TiaGo e dell'end effector del FR3 nello spazio cartesiano . . .	80
4.17	Campi Potenziali repulsivi	80

4.18	Confronto tra le velocità $ v_{avanz} $ delle sfere solidali al robot FR3 e la curva \mathbf{v}_{norm}	81
4.19	Confronto tra le velocità $ v_{avanz} $ del TiaGo e la curva \mathbf{v}_{norm}	81
1	Dati di input nel file Excel (parte 1)	88
2	Dati di input nel file Excel (parte 2)	89
3	Dati di input nel file Excel (parte 3)	89
4	Dati di input nel file Excel (parte 4)	90
5	Dati di input nel file Excel (parte 5)	90
6	Sequenza di smontaggio della batteria di un Audi Q5	91

Elenco delle tabelle

1.1	Scheda tecnica robot FR3	11
1.2	Scheda tecnica TiaGo Base	16
2.1	Correlazione tra componenti della batteria della Chevrolet Bolt 2017 e i rispettivi task	32
3.1	Relazione tra ciascuna sfera e il link del robot FR3 al quale sono solidali .	65

Capitolo 1

Introduzione

Un robot collaborativo è un robot concepito per interagire fisicamente con l'uomo in un dato ambiente di lavoro. Ciò trasmette un contrasto della maggior parte dei robot industriali, i quali erano progettati per operare in maniera autonoma o con una guida limitata e protetti da barriere. Essi non sostituiscono, bensì affiancano gli operatori, interagendo in modo funzionale all'esercizio di un compito: la collaborazione deriva dal loro utilizzo, oltre che dalle loro proprietà.

Con una definizione più generica, con "cobot" si intende l'insieme del "braccio", dell'Unità di Controllo e del Terminale di Programmazione.

Un braccio cobot è costituito da una base, fissa e ancorata alla piastra-base o piastra di fissaggio: un cobot può essere installato a pavimento, a parete, su piano inclinato, a soffitto ma sempre su una struttura di supporto tra la piastra-base e il piano. Esistono anche modelli "da tavolo". Dentro la piastra-base si trovano i connettori dei segnali (encoder, motori, freni) e il connettore Ethernet per la gestione dei dati che sale lungo tutta la struttura meccanica. Sopra la piastra-base, le parti del cobot vengono dette: colonna, braccio, avambraccio, corpo del polso, flangia, polso/testa. Il polso/testa è personalizzabile con le diversi utensili, tra i quali gripper, pinze e screwdriver.

I connettori dentro la piastra-base, talvolta posizionati anche dentro il corpo del polso, vengono collegati tramite appositi fili all'Unità di Controllo, una "centralina di elaborazione" esterna che monitora e gestisce il movimento degli assi attraverso appositi motori e un encoder, cioè un codificatore, ad alta risoluzione. L'Unità di Controllo si interfaccia sia con il "braccio" che con il Terminale di Programmazione da cui ha ricevuto le istruzioni. Queste non vengono impartite prima dell'inserimento sulla linea di produzione, ma sulla base dei movimenti e delle operazioni compiuti dall'operatore umano: il cobot, grazie ai sistemi di visione e di elaborazione dei dati provenienti dai diversi tipi di sensori, riesce a "vedere" l'operatore e ad adeguarsi.

La realizzazione del primo cobot risale al 1996 per mano di J. Edward Colgate e Michael Peshkin, due professori alla Northwestern University [12].

Un brevetto statunitense, depositato nel 1997, descrive i cobot come "un apparato e un metodo per l'interazione fisica diretta tra una persona e un manipolatore controllato da un computer". I primi risultati furono soddisfacenti dal punto di vista della sicurezza, ma molto limitanti in fatto di produttività. I cobot non avevano motori al loro interno

e il compito era quello di consentire il controllo del movimento da parte del computer, re-indirizzando o guidando il carico, in modo cooperativo con il lavoratore umano. Con il tempo questo problema si è risolto e anche i cobot hanno potuto operare in totale autonomia e collaborare effettivamente con l'uomo nello stesso spazio di lavoro grazie all'inserimento di motori al loro interno [12].

I robot collaborativi si sono affacciati sul mercato dell'automazione ormai 13 anni fa e, nonostante la loro diffusione sia stata piuttosto rapida e vasta, permangono ancora diverse incertezze circa la loro reale funzione e i vantaggi che comportano. Così come rimangono dubbi sui processi che i robot collaborativi sono in grado di automatizzare e i settori in cui possono venir integrati.

1.1 Caratteristiche peculiari

I robot si distinguono per dimensioni, portata, velocità, minore o maggiore flessibilità operativa e applicativa, costo, necessità o meno di barriere di sicurezza e per molte altre caratteristiche. In particolare, le differenze fra i robot tradizionali e i robot collaborativi possono essere descritte - semplificando - secondo le seguenti proprietà:

- Sicurezza
- Flessibilità
- Velocità di messa in opera

La maggior parte dei robot collaborativi possono operare senza l'ausilio di barriere di protezione intorno, a differenza dei robot tradizionali che invece necessitano di costose recinzioni a separare lo spazio dove operano da quello in cui agiscono le persone. Non è solamente questione di dimensioni e velocità (che sono tendenzialmente più contenute rispetto a quelle di un robot tradizionale) quanto di una diversa concezione della funzione del robot. Con un cobot è possibile impostare distanza e tempo di arresto e molte altre variabili che rendono la sua implementazione del tutto sicura anche in un'area affollata. I robot collaborativi inoltre offrono la possibilità di essere equipaggiati con un'ampia gamma di sensori che ne accrescono la sicurezza, come laser scanner e sistemi di visione che registrano la presenza dell'operatore controllando movimenti e tempi di reazione del cobot.

La sicurezza rappresenta soltanto il primo passo nel definire la reale capacità di un robot di cooperare con l'uomo. Altri elementi concorrono a definire questo parametro.

Un robot industriale tradizionale risulta performante solo su grandi volumi produttivi. Assicura infatti grande velocità e ripetibilità, su lotti lunghi e senza variazioni di prodotto. L'automazione cui un robot tradizionale dà origine è rigida. La flessibilità è invece la vera cifra stilistica che contraddistingue un robot collaborativo. Un cobot è piccolo e leggero: può quindi essere spostato agevolmente all'interno del layout industriale e applicato dove serve. Inoltre, la sua versatilità lo rende applicabile in molteplici operazioni: si tratta di un braccio capace di offrire il suo contributo operativo ovunque serva. Le dimensioni e il peso contenuti (circa 30 kg) non sono però l'unica caratteristica che rende i cobot facilmente dislocabili su attività di volta in volta diverse. Anche la possibilità di operare

in sicurezza senza recinzioni li svincola dall'averne una posizione fissa all'interno del layout. La facilità di programmazione è un'altra caratteristica che enfatizza la flessibilità dei cobot [30]. La programmazione di un braccio robotico è molto semplice per le operazioni più elementari e diffuse e può avvenire in una duplice maniera:

- Teach Pendant:: il tablet di programmazione con funzionalità touch screen contiene un template grafico che consente di impostare rapidamente i programmi del cobot.
- Free Drive: è possibile programmare il robot muovendo il braccio nello spazio e registrando dei waypoint specifici che forniranno al robot le coordinate attraverso cui ripetere il movimento programmato.

1.2 Applicazioni

I cobot sono in grado di automatizzare un gran numero di applicazioni. La semplicità e rapidità di programmazione, ne fanno strumenti flessibili pronti ad entrare in produzione e passare da un'attività ad un'altra in tempi molto rapidi [30]. Ecco alcuni esempi di cosa è possibile automatizzare con un robot collaborativo:

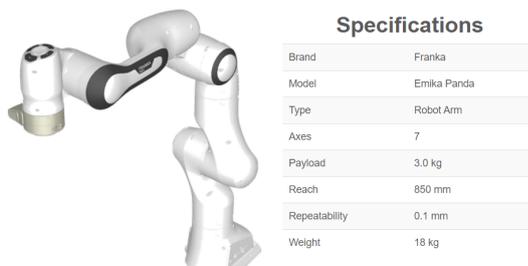
- Pick & Place: un must per i cobot. Precisi, sicuri, con elevatissimi standard di ripetibilità, sono la soluzione perfetta per sollevare gli operatori da operazioni ripetitive e tediose, generando al contempo efficienza e maggiore produttività
- Asservimento macchine utensile: grazie alla programmazione semplice e intuitiva possono essere facilmente implementati in applicazioni di asservimento macchine utensili come CNC o macchine per lo stampaggio plastico, anche per la produzione di prototipi e piccoli lotti di prodotto, così come automatizzare e ottimizzare la produzione di lotti lunghi
- Manipolazione materiale: è possibile effettuare con i cobot qualsiasi tipo di packaging e pallettizzazione. La semplicità di programmazione consente di impostare pattern personalizzati sia su pallet che all'interno delle confezioni
- Controllo qualità: anche per questa applicazione precisione e ripetibilità espresse dai cobot sono particolarmente performanti e indicate. Per questa fase strategica del processo produttivo i cobot sono un'automazione vantaggiosa in grado di generare efficienza e produttività. La vasta gamma di applicazioni come sensori e sistemi di visione 2D e 3D li rende applicabili in fasi come il controllo metrologico e dimensionale
- Assemblaggio: dotati di un sensore forza/coppia sul polso possono essere integrati in qualsiasi applicazione di assemblaggio garantendo costantemente l'erogazione della giusta forza nella manipolazione dei componenti. UR3e – il modello più piccolo della gamma UR e-Series – è dotato dell'infinite spin al polso, espressamente studiato per l'ottimale automazione delle operazioni di avvitatura

- Finitura superficiale: trovano applicazione anche in rapporto a materiali delicati come il legno e su superfici curve. Il sensore di forza sul sesto asse garantisce il rispetto delle superfici e ottime prestazioni
- Saldatura. I cobot possono automatizzare numerosi tipi di saldatura: TIG, MIG, plasma cutting, a punti, brasatura. Con i cobot è possibile incrementare qualità e precisione dei cordoni di saldatura e migliorare l'ergonomia del task sollevando gli operatori da un'operazione potenzialmente rischiosa e insalubre
- Erogazione: con un cobot è possibile impostare traiettorie di precisione micrometrica nell'applicazione di collanti e sigillanti. Questo da un lato aumenta la precisione e la coerenza del task, dall'altro riduce scarti e consumo di materiale.

1.3 Franka Emika FR3

FR3 è un robot collaborativo con 7 gradi di libertà, sviluppato dall'azienda tedesca FRANKA EMIKA. Questo robot, sviluppato per la ricerca, consente il controllo diretto e offre la possibilità di programmare e collegare sensori esterni (pacchetti e librerie per C++, ROS e MoveIt!). I sensori di coppia presenti sui sette assi rendono il braccio robotico FR3 estremamente sensibile, tanto che si arresta anche davanti ad un palloncino. Inoltre, questi sensori gli conferiscono un altissimo livello di precisione e stabilità. La pinza appositamente sviluppata è in grado di sollevare fino a 3 kg con una forza continua di 70 N (e una forza massima di 140 N). L'apertura massima della pinza è di 80 mm. Questo consente di sollevare e spostare un'ampia varietà di oggetti. Per rendere possibile tutto ciò, il produttore ha tentato di riprodurre la sensibilità e la destrezza di un braccio umano.

Figura 1.1: Robot Franka Emika FR3



L'installazione dell'FR3 è molto semplice e richiede soltanto pochi minuti. L'interfaccia utente del robot, "Desk", può essere consultata tramite un semplice browser Internet. Il braccio robotico è versatile e può essere utilizzato immediatamente per un'ampia varietà di applicazioni, tra cui: linee di produzione, ospedali e strutture sanitarie, laboratori, piattaforme di logistica, e così via. Si prega di notare che le diverse funzioni del robot implementabili in ambienti industriali (confezionamento, assemblaggio automatico, prova) sono incluse nel pacchetto applicativo fornito separatamente. FR3 può essere utilizzato immediatamente negli ambienti citati sopra ed installato senza un'approfondita analisi dei

rischi. Usando sistemi di cloud computing è anche possibile far eseguire compiti complessi ad una serie di robot FR3. Il braccio robotico FR3 combina precisione, l'integrazione di

Category	Specification
Arm	
Degrees of Freedom	7 DOF
Payload	3 kg
Sensitivity	Torque sensors in all 7 axes
Maximum Reach	855 mm
Joint Position Limits	A1: -166/166, A2: -101/101, A3: -166/166, A4: -176/-4, A5: -166/166, A6: -1/215, A7: -166/166
Joint Velocity Limits	A1: 150, A2: 150, A3: 150, A4: 150, A5: 180, A6: 180, A7: 180
Cartesian Velocity Limits	Up to 2 m/s end effector speed
Repeatability	+/- 0.1 mm (ISO 9283)
Interfaces	Ethernet, input for external enabling device, input for external activation device, Control connector, Hand connector
Interaction	Enabling and guiding button, selection of guiding mode, Pilot user interface
Mounting Flange	DIN ISO 9409-1-A50
Installation Position	Upright
Weight	~18 kg
Protection Rating	IP30
Ambient Temperature	+15°C to 25°C (typical), +5°C to +45°C (extended)
Air Humidity	20% to 80% non-condensing
Control	
Interfaces	Ethernet, power connector, Arm connector
Controller Size (19")	355 x 483 x 89 mm (D x W x H)
Supply Voltage	100 VAC - 240 VAC
Mains Frequency	47-63 Hz
Power Consumption	Max. 600 W, Average ~300 W
Active PFC	Yes
Weight	~7 kg
Protection Rating	IP20
Ambient Temperature	+15°C to 25°C (typical), +5°C to +45°C (extended)
Air Humidity	20% to 80% non-condensing
Hand	
Parallel Gripper	With exchangeable fingers
Grasping Force	Continuous: 70 N, Maximum: 140 N
Travel (Speed)	80 mm (50 mm/s per finger)
Weight	~0.7 kg
Desk	
Platform	Via modern web browser
Programming	Visual & intuitive, dialog-based
Apps	Composed into complex workflows

Tabella 1.1: Scheda tecnica robot FR3

C++ e ROS e maneggevolezza. Per questi motivi, è ampiamente utilizzato anche nelle università e nei laboratori di ricerca. Con il pacchetto Research App è possibile anche espandere le aree di applicazione del robot. Questo pacchetto di applicazioni opzionale, denominato "Research", è stato appositamente progettato per il settore dell'istruzione e della formazione. Tali applicazioni possono essere gestite tramite l'interfaccia utente "Desk" di FRANKA EMIKA [27].

1.4 Robot Mobili

Un robot mobile è una macchina automatica con la capacità di muoversi nell'ambiente e non sono fissati in una posizione fisica. I robot mobili possono essere "autonomi" (AMR - robot mobili autonomi), il che significa che sono in grado di spostarsi in un ambiente non controllato senza la necessità di dispositivi di guida fisici o elettromeccanici. In alternativa, i robot mobili possono fare affidamento su dispositivi di guida che consentono loro di percorrere un percorso di navigazione predefinito in uno spazio relativamente controllato. Al contrario, i robot industriali sono solitamente più o meno stazionari, costituiti da un braccio articolato (manipolatore multi-collegato) e un end effector fissato a una superficie

fissa [1].

I robot mobili sono diventati più comuni negli ambienti commerciali e industriali. Gli ospedali utilizzano da molti anni robot mobili autonomi per spostare i materiali [26]. I magazzini hanno installato sistemi robotici mobili per spostare in modo efficiente i materiali dagli scaffali di stoccaggio alle zone di evasione degli ordini. I robot mobili sono anche uno degli obiettivi principali della ricerca attuale e quasi tutte le principali università hanno uno o più laboratori che si concentrano sulla ricerca sui robot mobili. I robot mobili si trovano anche in contesti industriali, militari e di sicurezza [20].

I componenti di un robot mobile sono un controller, sensori, attuatori e un sistema di alimentazione. Il controller è generalmente un microprocessore, un microcontrollore incorporato o un personal computer (PC). I sensori utilizzati dipendono dai requisiti del robot. I requisiti potrebbero riguardare la navigazione stimata, il rilevamento tattile e di prossimità, la triangolazione, la prevenzione delle collisioni, la localizzazione della posizione e altre applicazioni specifiche. Gli attuatori di solito si riferiscono ai motori che muovono il robot che può essere dotato di ruote o gambe. Per alimentare un robot mobile solitamente utilizziamo l'alimentazione CC (con l'uso di una batteria) anziché CA [5].

1.5 Tiago Base

Il TiaGo Base è la parte inferiore del robot TiaGo, un robot di servizio mobile progettato dalla PAL Robotics per funzionare in ambienti interni e dotato di busto estensibile e braccio manipolatore per afferrare strumenti e oggetti. Il robot è in grado di trasportare carichi per un peso massimo di 50 kg. Il corpo principale del robot è formato da un cilindro di diametro pari a 54 cm e altezza pari a 30 cm. Il movimento è consentito da 6 ruote, di cui 2 di dimensione maggiore e 4 di dimensione minore. In esso sono inoltre integrati dei sensori di distanza basati su ultrasuoni per il rilevamento di ostacoli. La velocità di avanzamento massima del robot è 1 m/s.

Figura 1.2: AMR TiaGo Base



Queste caratteristiche permettono a questo robot mobile di poter operare in prossimità di una o più persone. Come effettuato in uno studio, il robot è equipaggiabile con elementi esterni come un microfono, un display LED e anche un cargo box [17]. Altri elementi con cui il TiaGo Base può essere integrato sono prodotti dalla Pal Robotics e in base ad esso, il TiaGo Base viene chiamato in un altro modo. Le varianti più comuni sono le seguenti:

- TiaGo Delivery
- TiaGo Actuated Conveyor
- TiaGo Static Roller
- TiaGo Shelf
- TiaGo Stackable Boxes

A differenza del busto e del braccio, il TiaGo Base può operare indipendentemente dagli altri componenti e può lavorare da solo o essere integrato anche con altri elementi non progettati per tale scopo. Ciò è possibile grazie al gran numero di porte accessibili dalla parte superiore del robot, coperte da una lastra di metallo [4]. Tali porte permettono anche di montare un supporto su cui montare l'elemento di interesse, che arriverà ad un'altezza maggiore. Questa opzione permette all'elemento montato sopra il TiaGo di interagire con oggetti posti ad altezze non normalmente raggiungibili (Es. oggetti poggiati sopra un tavolo).

Figura 1.3: Varianti del TiaGo in base agli elementi montati sopra



1.6 Electric Vehicle Battery

Vero cuore dell'auto elettrica, la batteria è l'elemento più importante tra quelli che costituiscono un'auto elettrica. Non è soltanto la componente in grado di garantire l'autonomia alla vettura, ma è anche quello che ne determina le prestazioni. Su un'elettrica, infatti, la performance del modello non è determinata tanto dalla potenza del motore, quanto dalla capacità dell'accumulatore. Più questo è grande e maggiore sarà l'autonomia e superiori

saranno le prestazioni. La batteria è vista come una sorta di serbatoio in cui immagazzinare l'energia necessaria a far muovere l'auto, ma in realtà è molto più di questo. Anche se la spinta arriva dal motore, le prestazioni di quest'ultimo sono influenzate direttamente dalla batteria e dalla sua capacità di erogare l'energia. Quindi, si può dire che in realtà il vero "motore" dell'auto elettrica sia proprio la batteria.

In ricarica accade il contrario: il flusso energetico in ingresso nella batteria si muove dal catodo verso l'anodo "restituendogli" gli elettroni. Questo processo avviene in ogni singola cella. Le celle sono i componenti di base della batteria: un certo numero di celle forma un modulo, e dall'unione di più moduli hanno origine le diverse batterie, la cui capacità totale è il risultato delle somme di quelle individuali delle celle.

In ricarica accade il contrario: il flusso energetico in ingresso nella batteria si muove dal catodo verso l'anodo "restituendogli" gli elettroni. Questo processo avviene in ogni singola cella. Le celle sono i componenti di base della batteria: un certo numero di celle forma un modulo, e dall'unione di più moduli hanno origine le diverse batterie, la cui capacità totale è il risultato delle somme di quelle individuali delle celle.

Il primo dato che si prede in esame per valutare le prestazioni di una batteria è la sua capacità, ossia la quantità di energia che può immagazzinare espressa in kWh (chilowattora), ma contano anche il modo e la rapidità con cui la immagazzina e soprattutto la eroga al motore. Queste caratteristiche sono espresse da voltaggio e amperaggio. Nello specifico:

- Capacità: viene espressa in kWh, che corrispondono alla quantità di kW (l'unità di misura della potenza, non solo elettrica) che la batteria è in grado di trasmettere oppure immagazzinare in un'ora. Per fare un esempio, una batteria da 50 kWh può far funzionare un motore da 50 kW alla massima potenza per un'ora, o uno da 100 kW per mezz'ora. Allo stesso modo, con un impianto di carica da 50 kW servirà un'ora per una ricarica mentre con uno da 100 kW basterà soltanto mezz'ora. Il kWh è usato anche come base per calcolare il consumo, e quindi l'efficienza, di un veicolo elettrico, espresso in km/kWh oppure in kWh/100 km
- Amperaggio: espresso in Ampere, è la quantità di energia che può essere trasmessa in un secondo
- Voltaggio: espresso in Volt, corrisponde alla forza, dunque alla rapidità, con cui una quantità di energia aprì a 1 Ampere viene trasmessa

Dunque, è facile comprendere come all'aumentare di voltaggio e amperaggio la batteria risulti più potente (anche in ricarica) perché il flusso di corrente che entra o esce sarà maggiore per volume e intensità.

I tipi di batterie si distinguono per la loro composizione chimica, ossia dai materiali, principalmente metalli per anodo e catodo ma anche i fluidi dell'elettrolita, che le compongono. Le batterie più tradizionali erano a base di piombo, poi sostituite da quelle al Nichel idruro metallo (NiMh) e poi da quelle agli ioni di Litio.

Le batterie agli ioni di litio sono oggi le più diffuse. Sono costituite da un anodo in litio e da un catodo in carbonio e devono il loro successo alla loro elevata densità: ciò significa che a fronte di dimensioni relativamente contenute sono in grado di generare grandi quantità di energia. Inoltre, non soffrono dell'effetto memoria del quale parlavamo nel capitolo

precedente, permettendo così cicli di carica e scarica anche parziali. Di contro, il ciclo di vita di queste batterie è abbastanza breve. Dopo circa 5 anni, infatti, la loro capacità di immagazzinare e cedere energia comincia a calare. Sotto i -10 gradi oppure oltre i 30 gradi, le batterie agli ioni di litio tendono a perdere efficienza. Il decadimento delle batterie dipende dall'usura del rivestimento di anodo e catodo (soprattutto del primo) che con il tempo riducono la capacità di accumulo. Questo non compromette le prestazioni come la potenza erogata, che rimangono costanti nel tempo, ma soltanto la capacità.

Le batterie allo stato solido rappresentano il prossimo step evolutivo delle batterie per automobili elettriche. Sono molti i costruttori che stanno lavorando allo sviluppo di questa nuova tecnologia, Tesla su tutti. A caratterizzare questa tipologia di accumulatore è l'elettrolita che qui è costituito da una sostanza solida e non più liquida. Questo incrementa enormemente la densità della batteria, assicurando prestazioni superiori del 50% rispetto agli ioni di litio, a fronte di dimensioni ancora più contenute. Inoltre, anche i tempi di ricarica scendono notevolmente [28].

Feature	Specification
Payload	100 kg
Dimensions / Max. speed	54 cm diameter / 30 cm height / 1.5 m/s
Traversable gap / step	150 mm
Motorized wheels	2
Caster wheels	4
Shock absorber	2
LEDs	4 individual
Audio Expansion Power	1x 5 W audio speaker
Expansion	10x GPIO (5x IN / 5x OUT) / additional emergency stop
Power	Battery / +12V/5A / +5V/5A
Wi-Fi	802.11 a/b/g/n/ac
Bluetooth	Smart 4.0 Smart Ready
Ethernet	2x Gigabit
USB	2x USB3 / 4x USB2
CPU	Intel i5
SSD	256 GB
Battery	1x36 V 20 Ah Li-Ion
Laser	Hokuyo URG-04LX-UG01
IMU	6 DoF
Differential platform	Integrated
Speaker	Integrated
User panel	Integrated
Indication lights	Integrated
Power connector	Integrated
On/Off button	Integrated
Emergency stop	Integrated
Electric switch	Integrated
Power charger	Integrated
Tactile control display	Integrated
Ubuntu LTS + ROS	Integrated
Navigation software	Integrated
Sonars	Integrated
CPU	i7 Haswell
RAM	16 GB
SSD	512 GB
RGB on top	Orbbec Astra
Additional battery	36V 20Ah Li-Ion
Laser	Sick TiM561 (10m), TiM 571 (25m)
2x channel CAN Board	Integrated
Dock Station	Integrated

Tabella 1.2: Scheda tecnica TiaGo Base

Capitolo 2

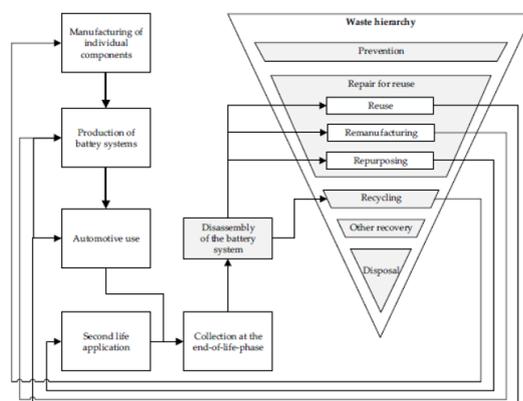
Analisi smontaggio batteria

2.1 Introduzione al problema

Nel capitolo precedente si è presentato come sono strutturati i robot collaborativi, con uno sguardo al robot antropomorfo Franka FR3 e TiaGo, e si è presentato com'è fatta la batteria di un'auto elettrica, quali sono le sue proprietà principali e quali sono le applicazioni. Ciò su cui non ci si è soffermati è il mondo che si cela dietro il tema dello smontaggio batteria: con l'aumentare del numero di auto elettriche nei prossimi anni, aumenterà anche il numero di auto da sostituire. Un'auto elettrica però, anche se impossibilitata a operare, ha comunque al suo interno materiali di grande valore e molto richiesti a livello industriale, come il litio e il rame. Si stima, infatti, che i due terzi dei costi di una batteria per auto elettriche derivino solo da questi materiali e dalla loro lavorazione dallo stato grezzo. Essendo questi elementi fondamentali nella costruzione delle comuni batterie, la loro richiesta è in continua crescita, rendendo la loro ricerca in natura non più sufficiente per soddisfare la domanda generale. Spesso però si fa affidamento su metodi inefficienti. In 2.1 è riportato i processi di recupero materiali, dal più semplice al più drastico:

Per far fronte a questo problema, oggi molte batterie, così come altri componenti elettronici simili, vengono sottoposte al processo di “recovery”. Il processo di recovery precede solo lo smaltimento, diventando quindi l'ultima spiaggia nel cercare di recuperare i materiali desiderati. Questo processo infatti richiede un gran consumo di soldi e di energia, che eccede di molto quello che si otterrebbe riciclando i materiali dopo averli rimossi. È evidente come riciclare questi materiali sia fondamentale per l'immediato futuro, ed è quindi necessario concentrarsi sul problema di smontaggio delle batterie di auto elettriche non limitandosi soltanto a rendere il processo fattibile e sicuro per l'operatore, ma andando anche alla ricerca di una soluzione ottimale, così da incrementare la produttività dell'intero processo di smontaggio.

Figura 2.1: Schema per il riutilizzo di una batteria di un veicolo elettrico



2.2 Tecniche per l'ottimizzazione dello smontaggio

Il processo di smontaggio di una batteria presenta numerose problematiche, che ancora oggi ostacolano la ricerca di una soluzione ottimale.

Uno di questi problemi è il design delle varie batterie. Essendo gli standard delle batterie ancora in fase di sviluppo, non esiste ancora una normativa che imponga ai costruttori di rispettare un determinato design. Ne consegue che le batterie vengono progettate esclusivamente in funzione del veicolo sul quale verrà montata. Il risultato di ciò è una moltitudine di modelli che, anche se con componenti simili, differiscono nella disposizione e nelle dimensioni, rendendo il processo di smontaggio differente [25]. Degli esempi di questo fenomeno sono la Tesla modello S, il cui pacco batteria è formato da 16 moduli, ciascuno composto da 444 celle cilindriche, oppure la Nissan Leaf il cui pacco batteria è composto da 48 moduli di dimensioni ridotte, ma composti da 4 celle ciascuno, di dimensioni maggiori rispetto ad una cella normale. Un altro esempio possibile è la BMW i3, con 8 moduli e 12 celle di grandi dimensioni per ciascun modulo[2].

La presenza di componenti simili in tutte le batterie (moduli batteria, collegamenti elettrici, componenti elettrici, supporti) rende comunque possibile ideare una sequenza preliminare con cui rimuovere ed eseguire lo smontaggio [23]:

- Rimozione del coperchio
- Separare i moduli batteria dagli altri componenti elettrici
- Scollegare meccanicamente i moduli batteria e i componenti elettrici dalla base della batteria
- Rimuovere i componenti elettrici
- Rimuovere i moduli batteria
- Smontare i singoli moduli batteria

Per lo smontaggio dei singoli moduli batteria è sufficiente ripetere gli stessi passaggi riportati sopra. Così come la batteria è costituita da un insieme di moduli, un modulo è infatti costituito da un insieme di celle.

Per lo smontaggio dei singoli moduli batteria è sufficiente ripetere gli stessi passaggi riportati sopra. Così come la batteria è costituita da un insieme di moduli, un modulo è infatti costituito da un insieme di celle.

Negli ultimi anni sono effettuati numerosi studi sulle batterie di alcune auto elettriche, alla ricerca di un metodo di smontaggio efficiente[23, 7]. Sono state analizzate due macrocategorie di batterie.

- Batteria di un’auto ibrida
- Batteria di un’auto elettrica plug-in

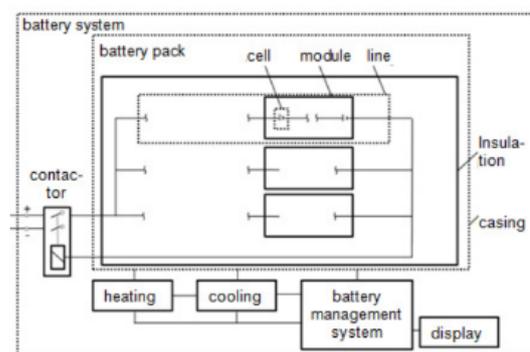
La differenza principale tra i due modelli è il contributo che danno alla propulsione: nell’auto ibrida, infatti, la batteria entra in funzione solo fino ad un determinato range di velocità, salvo poi cedere il ruolo di fonte d’energia al motore a combustione interna dell’auto. Questo metodo garantisce un miglioramento dell’efficienza e un minor impatto sull’ambiente ed è possibile guidare in città avvalendosi del solo motore elettrico, per poi servirsi del motore a combustione interna per le lunghe distanze [23]. Per l’auto full electric, invece, l’alimentazione è affidata solamente alla batteria.

Per ciascuna batteria è stato considerato un modello di automobile:

- Audi Q5 per la batteria di tipo ibrido
- Chevrolet Bolt 2017 per la batteria con plug-in

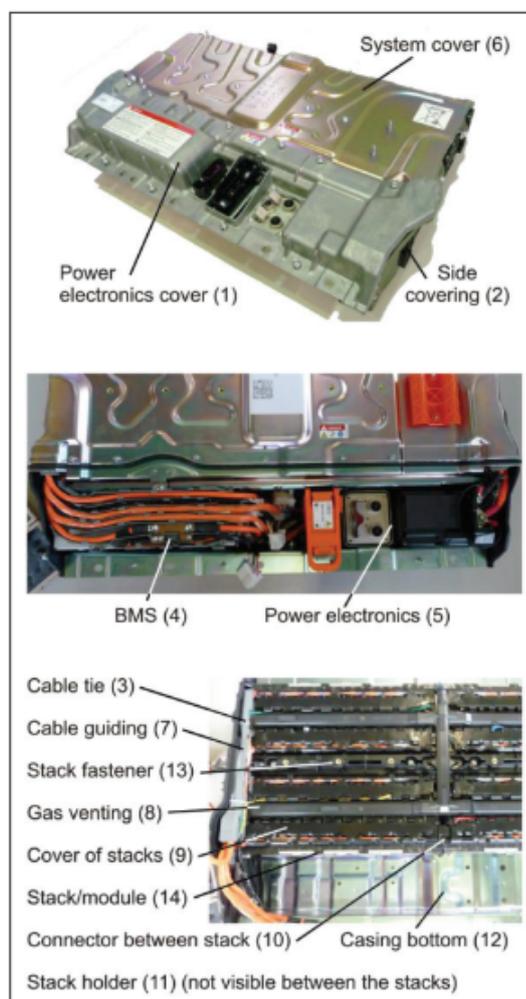
Con l’Audi Q5 si è dinanzi a un pacco batteria costituito da 4 moduli, un sistema di gestione batteria (BMS) e la relativa elettronica di potenza. Le dimensioni sono all’incirca 50 cm x 70 cm x 15 cm con un peso di circa 35 kg. La capacità si aggira intorno ai 5Ah e il voltaggio intorno ai 266 V [23].

Figura 2.2: Schema della batteria di un audi Q5



In 2.3 è invece riportato l’elenco dei componenti della batteria, opportunamente numerati:

Figura 2.3: Batteria di un Audi Q5, con i vari componenti numerati



Il metodo di smontaggio batteria applicato si è concentrato sull'estrazione dei moduli dal pacco batteria. Enumerate le parti principali, si è applicato il metodo della “matrice di smontaggio prioritario”. Con questo metodo si crea una matrice quadrata come quella 2.4.

Si nota come ogni componente della batteria è messo a confronto con gli altri. Il risultato di questo confronto è, per ogni casella della matrice, un numero che può essere pari a 0 o a 1. Ad ogni casella sono associati due componenti, un componente A sulla stessa riga e uno B sulla stessa colonna. Se il componente A deve essere smontato prima del componente B, la casella è riempita con l'1, altrimenti con lo 0. Sommando i numeri su ciascuna riga si definisce un possibile sequenza di smontaggio della batteria. In appendice si fornisce la sequenza di smontaggio risultante.

Per via della dimensione ridotta e del livello di complessità relativamente alto dei vari

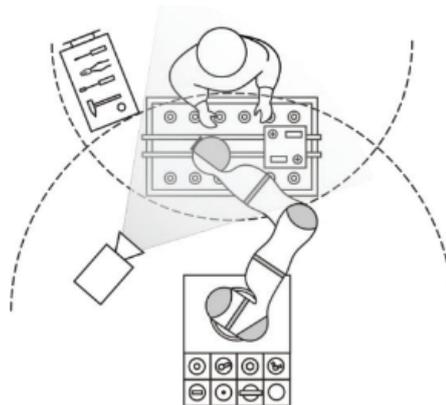
Figura 2.4: Matrice di smontaggio prioritario

Priority matrix dismantling	Successor														Total	Rank
	1	2	3	4	5	6	7	8	9	10	11	12	13	14		
1 power electronics cover	1	1	1	1	1	1	1	1	1	1	1	1	1	1	13	1
2 side covering	-1	1	1	1	0	1	1	1	1	1	1	1	1	1	11	2
3 cable tie	1	1	1	1	1	1	1	0	1	0	1	1	1	1	9	3
4 BMS	-1	-1	-1	0	1	1	1	1	1	1	1	1	1	1	9	3
5 power electronics cover	-1	-1	-1	0	1	1	1	1	1	1	1	1	1	1	9	3
6 system cover	-1	0	-1	-1	-1	1	1	1	1	1	1	1	1	1	8	6
7 cable guiding	-1	-1	-1	-1	-1	1	1	0	1	1	0	1	1	1	5	7
8 gas venting	-1	-1	-1	-1	-1	-1	0	0	1	1	1	1	1	1	4	8
9 cover of stacks	1	1	0	1	1	1	0	0	0	1	1	1	1	1	3	9
10 connector between stacks	-1	-1	-1	0	-1	0	0	0	0	1	1	1	1	1	3	9
11 stack fastener	-1	-1	0	-1	-1	0	-1	0	0	1	1	1	1	1	3	9
12 casing bottom	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	1	1	1	1	2	12
13 stack holder	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	13
14 modules/stacks	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	14

componenti, si è ritenuto che un solo operatore umano sia sufficiente per eseguire l'intero smontaggio.

Allo scopo di accelerare il processo sono stati pensati anche degli ambienti di smontaggio che includono, oltre a una persona, anche la presenza di un robot antropomorfo di supporto. Secondo questa applicazione, la persona si fa carico dei task più complessi, meno ripetitivi e in quantità minori, mentre al robot sono delegati i task più semplici e numerosi, ma che richiedono una grande ripetitività. In 2.5 è riportato uno schema possibile:

Figura 2.5: Stazione di lavoro con operatore e braccio meccanico fissato sul tavolo



In questa configurazione, uomo e robot condividono lo spazio di lavoro in cui giace la batteria. Ciascun operatore ha inoltre, nel proprio spazio di lavoro, un magazzino utensili

contenente tutti quelli che verranno utilizzati durante il processo di smontaggio batteria. Per l'applicazione in questione è richiesto un robot che sia allo stesso tempo leggero, così da ridurre i rischi associati ad eventuali urti, ma comunque con una certa capacità di carico per eseguire i task.

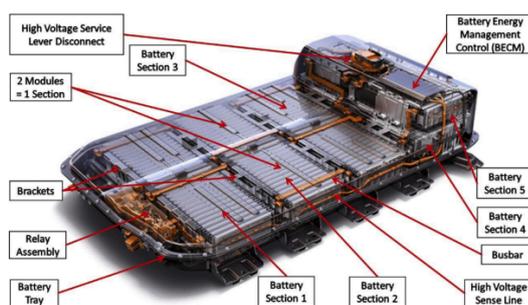
Nel caso della batteria della Chevrolet Bolt 2017 [7], si è dinanzi a una batteria di lunghezza pari a 1,6 m e di larghezza pari a 1 m e un'altezza complessiva di circa 20 cm. Le 288 celle di cui è composto l'intero pacco batteria sono suddivise nelle 5 sezioni così:

- Le sezioni che vanno dall'1 al 3 contengono lo stesso numero n_1 di celle
- Le sezioni 4 e 5 contengono lo stesso numero n_2 di celle, con n_2 minore di n_1 di alcune unità

Le sezioni 4 e 5 contengono lo stesso numero n_2 di celle, con n_2 minore di n_1 di alcune unità.

In Fig.14 è riportata l'immagine della batteria una volta rimosso il coperchio. Sono inoltre indicati gli elementi principali:

Figura 2.6: Interno di una batteria Chevrolet Bolt 2017



Si vede che ciascuna sezione è costituita da 2 moduli, per un totale di 10 moduli. Ciascuno dei due moduli di ogni coppia è identificato dal lato sul quale è posizionato, identificabile con Lato Guida e Lato Passeggero. Questa suddivisione è applicata anche alle busbar in arancione e ai supporti che fissano i moduli della batteria alla base.

Anche per questa batteria è stato concepito un metodo per ricavare una sequenza di smontaggio valida. In questo caso la batteria è stata suddivisa tra i suoi vari componenti, che sono stati classificati in due categorie: le parti (identificati dalla lettera P) e dispositivi di fissaggio (identificati dalla lettera F). Basandosi sui video di Kelly [31], sono state identificate tutte le parti e gli elementi di fissaggio smontati il vano batteria. In totale sono state identificate ed etichettate 76 parti e 374 elementi di fissaggio.

Le parti di maggior rilievo sono:

- Moduli batteria
- Busbar
- Supporti

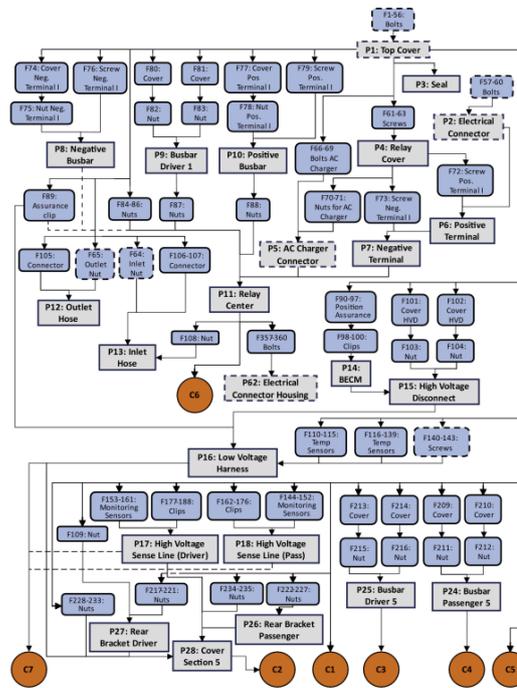
- Imbracature di alto voltaggio
- Imbracature di basso voltaggio

Gli elementi di fissaggio principali invece sono:

- Dadi
- Bulloni
- Viti

Tutti questi componenti sono collegati tra loro da una sequenza di smontaggio predefinita, dove ogni componente è preceduto da quei componenti che devono essere necessariamente rimossi prima, ed è seguito da altri componenti che devono essere rimossi necessariamente dopo. In 2.7, 2.8 e 2.9 è riportata questa gerarchia, espressa come diagramma di flusso:

Figura 2.7: Diagramma di smontaggio della batteria della Chevrolet Bolt 2017 (parte 1)



i criteri per il TAA per il modello di punteggio e ciascuno è stato ponderato in modo diverso a seconda dell’importanza di ciascun criterio. In 2.10 sono riportati i 5 criteri più rilevanti per ciascun metodo:

Figura 2.10: Criteri per la valutazione dell’assegnazione dei task ai due operatori (uomo e robot)

Category	Criterion Number	Criterion Description
NA	1	Number of Motions (human)
	2	Duration of manual disassembly time in seconds
	3	Danger (High voltage protection, hazardous materials)
	4	Weight
	5	Priority (value)
TAA	1	Complexity of motion (for robot, number of different motions)
	2	Access for end effector
	3	Possible detection
	4	Automation potential for robotic end effector
	5	Material handling

Per ogni criterio, sia nel caso di NA che TAA, va assegnato un numero intero che va da -2 a 2 (compreso lo 0). Si hanno quindi 5 valori NA_i e 5 valori TAA_i da cui ricavare i valori complessivi.

I risultati di questo metodo hanno evidenziato come sia conveniente automatizzare solo le operazioni di movimentazione e presa ripetitive. Oltre che sulla Chevrolet Bolt 2017 è stato effettuato lo stesso studio anche sull’Audi Q5 e su altre batterie. In 2.11, 2.12, 2.13 e 2.14 sono riportati i risultati.

Con quest’analisi, è possibile suddividere i task in due insiemi: quello dei task difficilmente automatizzabili e quello e quello dei task facilmente automatizzabili. È quindi possibile aggiungere nel processo di smontaggio anche un robot collaborativo, al quale delegare le operazioni facilmente automatizzabili. In questo modo si hanno due operatori che, lavorando in parallelo, rendono il processo di smontaggio della batteria notevolmente più rapido.

Anche se l’analisi preliminare riportata sopra è essenzialmente corretta, presenta diverse lacune:

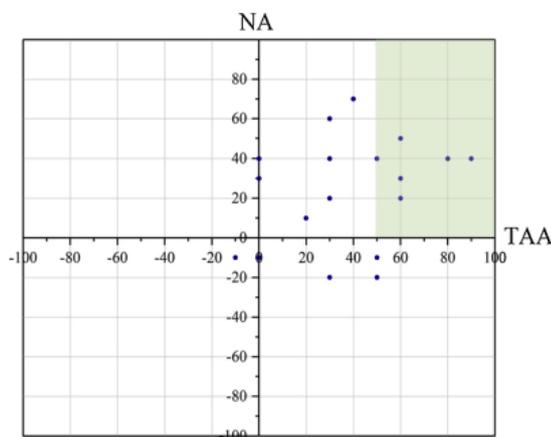
- La sequenza di smontaggio tiene conto esclusivamente conto della precedenza, lasciando libertà sull’ordine nel caso in cui due o più task possano essere eseguiti in contemporanea
- La sequenza di smontaggio non considera le possibili combinazioni che si possono andare a formare in presenza di due operatori che lavorano in contemporanea.

Inoltre, si nota che la predisposizione di una stazione di lavoro con un robot montato su un piano fisso in prossimità della batteria non è una soluzione applicabile in linea generale. Nel caso della batteria della Chevrolet Bolt 2017, infatti, l’ingombro stesso impedirebbe al robot di raggiungere qualunque zona della batteria, vincolando il robot ad agire in una zona molto limitata. Anche scegliere di delegare una data zona al robot e lasciar operare la persona in quelle rimanenti non è un’opzione contemplabile. In questo modo, infatti, si limita di molto l’impiego del robot collaborativo e al contempo non si sfrutta a pieno la

Figura 2.11: Valori NAI e TAAi riferiti ai componenti della batteria dell'Audi Q5

Step #	NA	TAA
Step I: Unscrew covers and casing bottom:	30	20
Step II: Removal of power electronics cover and side covering:	-10	50
Step III: Disassembly of the live lines from the modules/ stacks:	Not enough information to assess	Not enough information to assess
Step IV: Cutting of cable ties:	10	20
Step V: Disassembly of the plug connection between the cell controllers and the BMS:	-10	-10
Step VI: Removal of BMS and power electronics:	20	30
Step VII: Cutting of the bus for the thermo sensors:	-10	0
Step VIII: Disassembly and removal of system cover:	-20	50
Step IX: Unscrew and removing of cable guiding:	40	50
Step X: Removal of gas venting and the cover of the stacks:	20	60
Step XI: Disassembly and removal of the connector between the stacks:	30	0
Step XII: Unscrew and removal of stack holders:	50	60
Step XIII: Removal of casing bottom:	-20	30
Step XIV: Unscrew and removal of stack fastener:	40	80
Step XV: Removal of stacks:	70	40
Steps for the disassembly of the battery modules:		
Step I: Unscrewing of nuts on the cell contacts	40	90
Step II: Removal of cables and cell connectors	40	0
Step III: Unscrewing and removal of the side covers:	30	60
Step IV: Removal of battery cells:	60	30

Figura 2.12: Valori NAI e TAAi riferiti ai componenti della batteria dell'Audi Q5 (grafico)

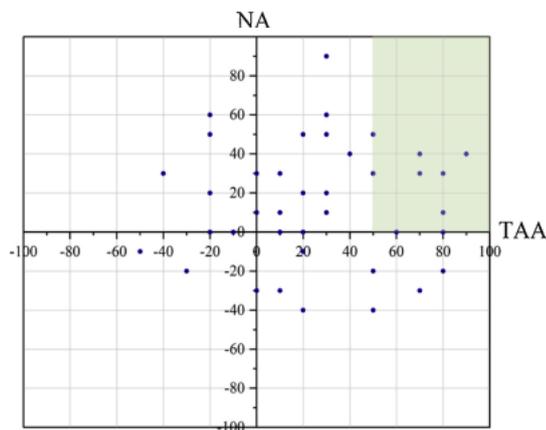


capacità del robot collaborativo di non urtare la persona. Come si vedrà successivamente, il problema viene affrontato montando il robot antropomorfo sopra una base mobile, nello specifico un AMV (Autonomous Mobile Vehicle), ossia un robot mobile capace di spostarsi nel piano, e quindi permettere al cobot di eseguire i task a prescindere da dove sono situati nello spazio di lavoro.

Figura 2.13: Valori NAI e TAAi riferiti ai componenti della batteria della Chevrolet Bolt 2017

Step	Involved Parts or Fasteners	Description	NA	TAA
D1	F1-56	Bolts for Top Cover	40	90
D2	P1	Lifting of Top Cover	10	30
D3	F57-60	Bolts for Electrical Connector	30	70
D4	P2	Electrical Connector	-20	50
D5	P3	Seal	-30	0
D6	F74, F77, F80, F81	Covers for Busbars (Front)	20	20
D7	F61-63, F66-69, F75-76, F78-79, F82-83	Nuts/ Screws for Busbars (Front) and Relay Cover	60	30
D8	P4, P8-10	Relay Cover, Busbars	30	50
D9	F70-73, F84-88	Nuts, Screws below Relay Cover	50	30
D10	P5-7, P11	Grabbing of Relay Center, Terminals, AC-Charger	30	-40
D11	F105-108	Connectors and Nut for Coolant Hoses	-10	-50
D12	F64-65	Big Nuts for Coolant Hoses	-20	-30
D13	P12, P13	Grabbing of Coolant Hoses	-40	50
D14	F357-360	Nuts for Electrical Connector	0	60
D15	P62	Electrical Connector	-40	20
D16	F89-100, F110-139,	Assurance Clips, Temp. Sensors, BECM clips	60	-20
D17	P14	BECM	0	80
D18	F101-102, F189-190, F193-194, F197-198, F201-202, F205-206 F209-210, F213-214	Covers for Busbars, High Voltage Disconnect	50	20
D19	F103-104, F109, F140-143, F191-192, F195-196, F199-200, F203-204, F207-208, F211-212, F215-280, F290-326	Nuts, Bolts and Screws for Busbars and Brackets, High Voltage Disconnect and Battery Sections	50	50
D20	P15	High Voltage Disconnect	0	60
D21	P16	Low Voltage Harness	0	-20
D22	F144-188	Clips and Temp. Sensors for HV Sense Lines	50	-20
D23	P17, P18	HV Sense Lines	20	-20
D24	P26-27	Rear Brackets	0	60
D25	P28	Cover Battery Section 5	20	30
D26	P19-25, P29-30, P32-35, P37-38, P51-52, P53-54	Busbars and Brackets	50	30
D27	P40-41	HT Mats Battery Section 5	10	10
D28	F284-289	Hose Champs	10	0
D29	P42-44	Hoses	-30	10
D30	F281-283	Nuts for Coolant Plate	30	80
D31	P45	Coolant Plate Section 5	30	10
D32	P46	Insulating Pad Battery Section 5	0	20
D33	P47	Cover Battery Section 4	20	30
D34	F327-328	Clips HV Harness Battery Section 4	0	-10
D35	P48-49	Side Brackets Battery Section 4	0	60
D36	F337-342	Bolts Battery Section 4	40	70
D37	F329-336	Retainers, Position Assurance Battery Section 4	30	0
D38	P55-56	HV Monitoring Circuit Battery Section 4	0	60
D39	P36, P39, P50, P57	Battery Sections 1-4 Lifting	90	30
D40	P63-70	Heat Transfer Mats	10	10
D41	F343-356, F361-371, F375	Bolts for Braces, Coolant Plate	10	80
D42	P58-61	Braces	-30	70
D43	F372-374	Bolts for Coolant Plate	-20	80
D44	P71	Coolant Plate	0	10
D45	P72-75	Insulating Pads	-10	20
D46	P76	Handling of Battery Tray	40	40

Figura 2.14: Valori NAI e TAAi riferiti ai componenti della batteria della Chevrolet Bolt 2017 (grafico)



2.3 Metodo di smontaggio con algoritmo di ottimizzazione

Nella sezione precedente sono stati illustrati dei metodi che aiutano a rendere la sequenza di smontaggio della batteria di un'auto elettrica più rapido ed efficiente. Questi metodi però si limitano a indicare la scala gerarchica presente tra i vari task oppure il criterio con cui ripartire i task tra operatori. Se si è infatti in presenza di due task dello stesso tipo (quindi stessi NA e TAA), ma senza che l'esecuzione di un task abbia ripercussioni sull'altro, è evidente come non forniscono indicazioni chiare sull'ordine di esecuzione dei task. Per contro, invece, l'esecuzione di uno o dell'altro può avere ripercussioni sul tempo complessivo di smontaggio.

Si tratta del tipico problema di ottimizzazione combinatoria, riassumibile dal "Problema del commesso viaggiatore". La base di questo problema è far sì che il commesso viaggiatore debba raggiungere un numero definito di città. Ipotizzando che le città siano tutte collegate tra loro, ci saranno più soluzioni al problema, ma ce ne sarà solo una che permette di raggiungere il risultato scegliendo il percorso più breve possibile [9].

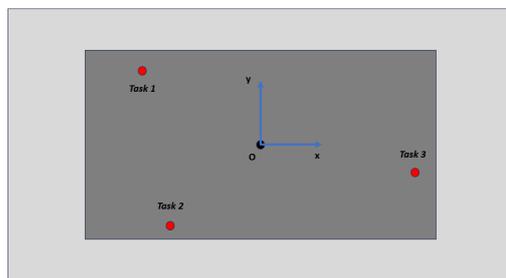
Applicando il problema ad una situazione che può verificarsi nello smontaggio batteria, si considera una situazione con tre task da dover eseguire, come in 2.15:

Ipotizzando che l'esecuzione di uno non cambi nulla per gli altri due, ne consegue che ci saranno sei possibili permutazioni con cui i task possono essere eseguiti. Se invece nel problema si considerano n task, il numero di sequenze di smontaggio possibili possono essere espresse con la seguente formula:

$$N_{\text{Seq Smontaggio}} = N_{\text{Permutazioni}} = \frac{n!}{(n-n)!} = n! \quad (2.1)$$

Dovendo scegliere la sequenza di smontaggio ottimale, si opta per quella che richiede il minor tempo per completare tutti i task. Se si ipotizza che il tempo ciclo per l'esecuzione di ciascun task non cambia indipendentemente dall'ordine con cui essi vengono eseguiti,

Figura 2.15: Posizione di 3 task generici nello ambiente di lavoro



ne consegue che l'unico fattore che influisce sul tempo ciclo totale sono le distanze coperte per spostare l'operatore nello spazio di lavoro. Queste distanze variano in base all'ordine con cui si decide di eseguire i task ed è il cuore del "Problema del commesso viaggiatore". L'ottimizzazione dello smontaggio batteria, però, pur avendo dei punti in comune con il problema del commesso viaggiatore, presenta anche delle differenze notevoli. Il primo, infatti, è influenzato da altri fattori:

- Il cambio utensile per l'esecuzione di task differenti
- La presenza di un robot antropomorfo come secondo operatore
- La ripartizione tra uomo e robot dei task da eseguire
- Il rallentamento del robot dovuto all'eventuale distanza ridotta tra uomo e robot

Nel problema di smontaggio della batteria, infatti, oltre a raggiungere la zona in cui eseguire il task, bisogna tener conto del tempo di esecuzione del task. In questo caso si può ipotizzare che ad ogni task è associato un utensile con cui eseguire il suddetto task. Ne consegue che, durante la sequenza di smontaggio, è possibile ritrovarsi in una situazione in cui l'operatore deve eseguire un task che richiede un utensile A, ma ha appena eseguito un task che necessitava di un utensile B. Come conseguenza di ciò, l'ordine di scelta dei task non va solo ad influenzare il tempo necessario per spostarsi da una zona dello spazio di lavoro ad un'altra, ma anche il numero di cambi utensile che avverranno durante l'intero processo di smontaggio.

La seconda differenza che rende il problema di ottimizzazione ancora più complicato è la presenza di un secondo operatore che contribuisce allo smontaggio. Trattando il caso come un "Problema del commesso viaggiatore", infatti, non ci si trova più dinanzi a un viaggiatore che deve raggiungere tutte le città nel percorso più breve possibile, ma si sta considerando che i commessi viaggiatori a lavoro sono due. In questa situazione ognuno dei due deve passare per una sola città e si deve calcolare il percorso che entrambi devono compiere in parallelo tale da minimizzare il maggiore tra i due.

Nel problema dello smontaggio batteria c'è anche la criticità di dover assegnare i task ad un operatore: come evidenziato dallo studio [23], non tutti i task possono essere eseguiti da entrambi gli operatori. Ci sono infatti task che sono delegabili esclusivamente all'uomo,

altri task che possono essere svolti indipendentemente da entrambi gli operatori e altri ancora che è preferibile delegare al robot collaborativo. In questo terzo caso, però, non va esclusa la possibilità che i task finiscano comunque per essere svolti dalla persona.

Infine, si deve considerare l'effetto che ha la distanza tra i due operatori nel corso dello smontaggio. In un caso in cui i task sono sparsi nell'intero spazio di lavoro, può capitare che, per un dato intervallo, i due operatori debbano eseguire task diversi, ma in zone molto vicine o addirittura nella stessa zona. Nella pratica, questo concetto implicherebbe che i due operatori debbano lavorare a distanza pressoché nulla o addirittura vedere i rispettivi corpi "compenetrarsi" (situazione fisicamente impossibile). Qualunque sia la situazione che si andrebbe a creare in linea teorica, nella realtà il risultato sarebbe lo stesso, ovvero un rallentamento del robot, che si adatta ai movimenti della persona. Seguendo le 3 leggi di Asimov, un robot non può danneggiare l'uomo, quindi, se un robot collaborativo si trova nelle vicinanze, deve essere in grado di mantenere una distanza di sicurezza e una velocità adeguata. Da ciò ne consegue che il robot potrà impiegare più tempo di quanto previsto in base a quanto i movimenti della persona interferiranno con i movimenti del robot FR3.

Si nota subito come il problema di ottimizzazione aggiornato con le nuove ipotesi si distacca non di poco dal problema di ottimizzazione classico. Nella situazione aggiornata, si è davanti a due operatori, un uomo e un robot antropomorfo, che devono svolgere un numero definito di task legati tra loro da un diagramma ad albero di forma generica da rispettare obbligatoriamente. A ciascun operatore viene assegnato un gruppo di task definito da eseguire in un ordine tale da ottimizzare il tempo ciclo complessivo dello smontaggio della batteria. Da qui il sorgere dell'ostacolo principale dietro la consegna: per ottimizzare il problema, i singoli aspetti sopra elencati dovranno essere ottimizzati nella loro totalità, impedendo quindi di suddividere il problema di ottimizzazione globale in problemi di ottimizzazione singoli. In aggiunta a ciò, anche le sequenze di smontaggio dei due operatori dipendono l'una dall'altra, e l'ottimizzare il tempo ciclo in una può ridurre, aumentare o lasciare immutato il tempo ciclo dell'altra. In un contesto in cui il tempo ciclo da ottimizzare è il massimo dei tempi ciclo delle sequenze di smontaggio singole, ci si può trovare in una situazione in cui si deve aumentare il tempo ciclo di un operatore per poter ridurre il tempo ciclo dell'altro.

Trovandosi di fronte a un problema di tale portata, algoritmi di ottimizzazione ispirati alla genetica [10], o agli sciami [24] sono stati scartati, in quanto troppo slegati dal problema in questione. Al loro posto è stato realizzato un algoritmo apposito, che restituisce dei risultati che si riveleranno accettabili. L'algoritmo è stato realizzato con il software Matlab. Per la validazione del codice era necessario conoscere il diagramma di smontaggio della batteria su cui testarlo; perciò, la validazione è stata effettuata sulla batteria della Chevrolet Bolt 2017 citata in precedenza.

Nella realizzazione di questo algoritmo si è tenuto conto dei seguenti aspetti:

- Il diagramma di smontaggio della batteria e della gerarchia tra i vari task
- La posizione dei componenti della batteria
- Le dimensioni delle aree di lavoro dei vari task, in funzione dell'operatore che esegue il task

- Il tempo ciclo ottimale con cui ogni operatore può svolgere un task. Essi dipendono dal tipo di task e dall'operatore che lo svolge
- Il tempo impiegato dagli operatori per spostarsi all'interno della zona di lavoro
- Il tempo impiegato da ciascun operatore per effettuare il cambio utensile
- Il tempo supplementare che gli operatori devono impiegare per svolgere dei task in seguito alla vicinanza uomo-robot antropomorfo
- I tempi morti in cui l'operatore non può eseguire task

Inoltre, sono state fatte le seguenti ipotesi semplificative:

- I tempi ciclo ottimali dei vari task sono ipotizzati
- I due operatori non possono operare sullo stesso task
- I tempi di cambio utensile sono supposti costanti
- I tempi aggiuntivi dovuti alla vicinanza uomo-robot dipendono dall'intervallo di tempo e dalla distanza uomo-robot. La formula con cui si calcolano è ricavata sperimentalmente
- I due operatori possono occupare la stessa zona nello spazio
- L'interferenza tra uomo e robot mobile è ignorata

Per i dati di input del problema si fa distinzione tra:

- Dati fissi, contenuti in una tabella Excel creata appositamente
- Dati variabili, inseriti nello script di Matlab

Nel file Excel sono stati inseriti i dati relativi alla batteria come:

- I nomi dei task
- La posizione nel piano rispetto al baricentro della batteria dei componenti da smontare in ciascun task
- L'area occupata dai componenti da smontare in ciascun task
- La ripartizione dei task tra i due operatori
- Gli utensili necessari ai due operatori in ciascun task (nel caso in cui la persona non necessita di un utensile non è stato indicato niente)
- I tempi ciclo ottimali per ciascun task
- La gerarchia di esecuzione dei task descritta dal diagramma ad albero espressa in forma tabellare

Gli altri input sono i tempi di cambio utensile di ciascun operatore. Per motivi pratici, ogni task è stato denominato con una o più lettere e un numero intero. La lettera dipende dai componenti che vengono rimossi nel task, mentre il numero è scelto arbitrariamente e serve per distinguere tra loro task simili. In 2.1 è riportata l'associazione tra componenti e lettera identificativa del task

Componente	Codice Task
AC charger	AC
Battery Section	BS
BECM	BM
Bolt Cover Section	B0
Bolts	B
Brace	BC
Bracket	BR
Busbar	BU
Clip harness	CH
Clips	CL
Connector	CN
Coolant plate	CP
Cover Section	CS
Covers	CO
Elect. connect.	EC
Heat Transf Mat.	HT
High Voltage	HV
Hose	H
Hose Clamps	HC
Hose nuts	HN
HV monitoring	HVM
IN Hose	IN
Insulating pad	IP
Monit. sensors	MS
Neg. Terminal	TN
Nuts	N
Out Hose	OH
Pos. Terminal	TP
Position Assurance	PA
Relay center	RC
Relay cover	RE
Screws	S
Seal	SL
Temp. sensors	TS

Tabella 2.1: Correlazione tra componenti della batteria della Chevrolet Bolt 2017 e i rispettivi task

Definiti i dati di input del problema, l’algoritmo restituisce:

- Le sequenze di task per ciascun operatore
- I tempi totali impiegati dai due operatori
- I tempi morti dei due operatori
- I tempi di cambio utensile
- I tempi utili di esecuzione dei task
- I tempi di smontaggio aggiuntivi dovuti all’interazione uomo robot

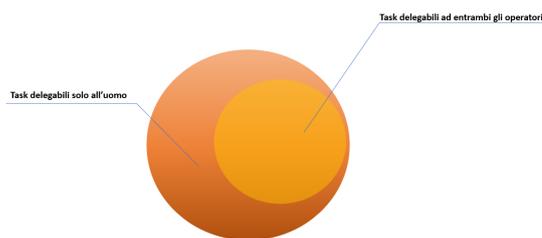
In appendice sono riportate le tabelle contenenti tutti i dati.

2.4 Spiegazione dell’algoritmo di ottimizzazione

L’algoritmo, dopo aver letto il file Excel di input e salvato i dati in opportune variabili, è costituito da un ciclo while con al suo interno una serie di if, che analizzano le diverse possibilità che si possono andare a creare. Partendo da un numero N_{tot} di task da effettuare, se si considera il ciclo j -esimo, ci si ritrova con un numero N di task ancora da eseguire. Questi si possono dividere in:

- Task delegabili all’uomo
- Task delegabili ad entrambi gli operatori

Figura 2.16: Insieme dei task delegabili all’uomo e insieme dei task delegabili a entrambi gli operatori



L’intersezione dei due insiemi corrisponde ai task effettuabili da entrambi gli operatori. È da considerare anche una seconda suddivisione:

- Task effettuabili
- Task non ancora effettuabili

Figura 2.17: Insieme dei task effettuabili subito e insieme dei task non ancora effettuabili

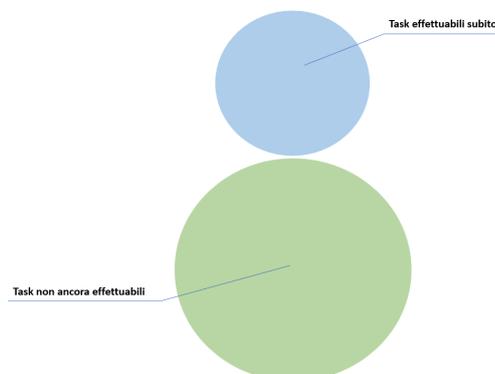
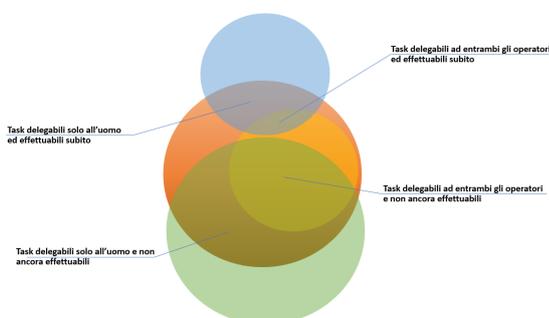


Figura 2.18: Combinazione degli insiemi rappresentati in 2.16 e 2.17



Per definizione, solo l'insieme dei task eseguibili va considerato, portando al sistema di insiemi e sottoinsiemi in 2.18

La costruzione delle sequenze di smontaggio avviene tramite la somma dei tempi impiegati per eseguire i singoli passaggi. Di volta in volta ci si ritrova con due tempi ciclo associati ai due operatori.

Si valuta quindi a quale operatore è associato il tempo ciclo minore, per poi concentrarsi solo su di esso e sui task che può svolgere. Per la scelta tra i task sono stati considerati due metodi:

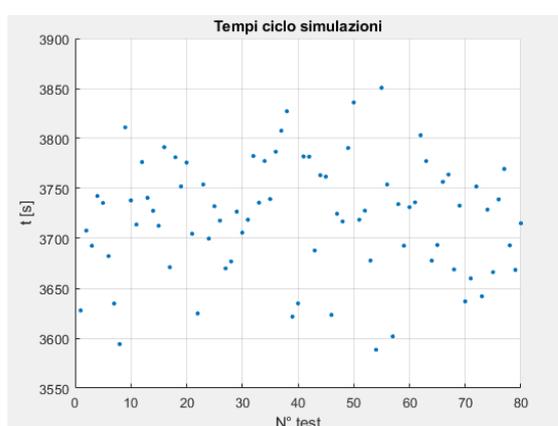
1. Analisi di ogni possibile permutazione
2. Scelta randomica tra i task disponibili

Inizialmente si è studiato il metodo 1: considerando un caso in cui un operatore può scegliere indistintamente quale task effettuare tra n effettuabili, si studia il problema con n casi, ciascuno che considera un nuovo task diverso. Di conseguenza, anche il tempo ciclo relativo all'operatore cambia da caso a caso.

Con questo metodo è possibile analizzare qualsiasi permutazione possibile, per poi scegliere quella che offre il tempo ciclo complessivo minore. Applicando questo metodo al diagramma ad albero della batteria della Chevrolet Bolt 2017 è emerso come le sequenze

di smontaggio possibili siano troppo elevate, al punto che neanche il software Matlab non riesce ad elaborare i dati in tempi accettabili. Per questo si è poi scelto il metodo 2. In questo caso, in presenza di più task da poter eseguire, la scelta è effettuata casualmente dall’algoritmo stesso. In questo modo l’output finale sarà una sola sequenza di smontaggio per ciascun operatore e il software restituirà un risultato quasi istantaneamente. Per rendere questo metodo applicabile si deve giungere al compromesso di non poter ottenere la sequenza di smontaggio ottimale, ma una accettabile. È infatti possibile far girare il codice per un numero indefinito di volte e, di volta in volta, confrontare il nuovo output con quello vecchio e scegliere quello che ottimizza meglio il problema. Il codice, testato per un numero sufficiente di volte ha restituito la serie di valori di 2.19

Figura 2.19: Plottaggio di 80 tempi di simulazione con il nuovo algoritmo di ottimizzazione



Confrontando i valori con quelli presentati nello studio [11], si nota che l’output medio che fornisce l’algoritmo fornisce dei tempi di smontaggio più che accettabili.

Scelto il nuovo task, vengono conteggiati:

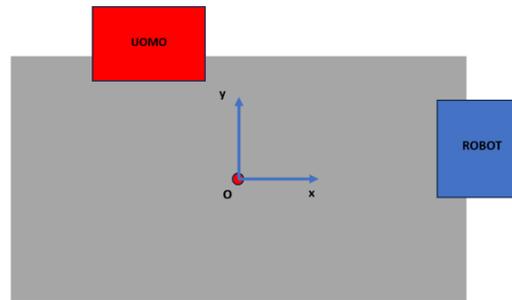
- l’eventuale tempo di cambio utensile
- l’eventuale tempo che l’operatore impiega per spostarsi nello spazio di lavoro
- Il tempo ciclo riportato nel file Excel

L’ultimo step vede il calcolo dei tempi dovuti all’interferenza tra i due operatori. Approssimati i due operatori come due parallelepipedi nel piano, come in 2.20.

Si calcola la distanza minima tra i due e, se questo valore scende al di sotto di una certa soglia, si calcola, per ciascun operatore, un apposito coefficiente inversamente proporzionale alla distanza. Noto questo coefficiente e il tempo che i due operatori trascorrono in questa situazione, è possibile determinare l’intervallo di tempo supplementare che i due operatori impiegheranno per svolgere i rispettivi task.

Notare che, se il problema cade in questo caso particolare, i tempi ciclo di entrambi gli operatori aumentano, evidenziando come il problema del calcolo del tempo ciclo è estremamente complesso, al punto che, in un caso reale, in cui i tempi ciclo ipotizzati cambiano di volta in volta, è impossibile riuscire a rispettare la sequenza di smontaggio. Per questi

Figura 2.20: Rappresentazione schematica dei due operatori nell'ambiente di lavoro



motivi, un eventuale interazione uomo robot prevista dalla sequenza di smontaggio teorica potrebbe non verificarsi nella realtà, invece un caso di interazione non previsto potrebbe invece avvenire. Per questi motivi, nello studio effettuato nel Capitolo 4, verranno prese in considerazione più sequenze di smontaggio aventi un tempo ciclo simile.

Capitolo 3

Analisi Interazione Uomo Robot

Per studiare il comportamento dei robot durante lo smontaggio della batteria si è effettuata una simulazione in ambiente 3D con il software Coppeliasim (ex V-REP). Il software è in grado di interfacciarsi con diversi linguaggi di programmazione (come C, C++, Python, Java, Lua, Matlab e Octave) usando un processo denominato “chiamata API”. Si definisce “chiamata API” un processo attraverso il quale due software scambiano dati. Quando un’applicazione fa una chiamata API, invia ad un’altra applicazione una richiesta specifica che viene ricevuta e processata da una API. Le API (Application Programming Interfaces) sono elementi di una struttura intermedia che permette a diverse applicazioni web di comunicare tra di loro. Nei software che ne dispongono, le API si occupano di esporre alcuni dati verso l’esterno, in modo che le informazioni possano essere condivise con le altre applicazioni che ne facciano richiesta.

L’ambiente in Coppeliasim è stato realizzato in scala 1:1 con i seguenti elementi:

- Il tavolo su cui è poggiata la batteria
- La batteria con i suoi componenti più importanti
- Un manichino che riproduce i movimenti della persona
- Il robot mobile TiaGo
- Il robot antropomorfo Franka Emika FR3
- Un rialzo per permettere al robot FR3 di operare sulla batteria
- Posa di partenza del TiaGo
- Posa di arrivo del TiaGo
- Punto di arrivo dell’end effector del FR3

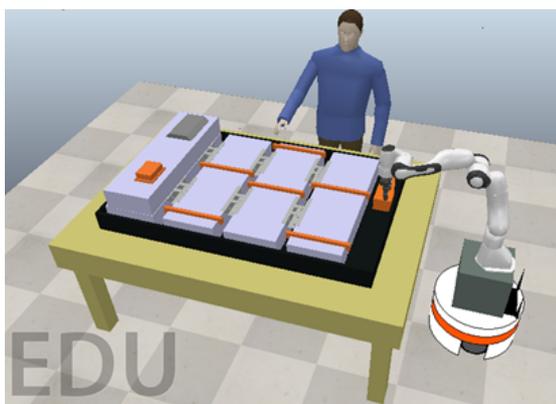


Figura 3.1: Ambiente di simulazione in Coppelasim con tutti gli elementi

In questo ambiente di simulazione vengono riprodotti i movimenti dei due robot e della persona ed è possibile osservare come i due operatori si muovono e verificare la completa assenza di urti tra i due operatori. L'obiettivo di questo lavoro, infatti, è verificare che i due operatori possano lavorare allo smontaggio di una batteria di un veicolo elettrico condividendo lo stesso spazio di lavoro. Tale situazione implica, in base a quanto spiegato nel Capitolo 1, che i due robot, per non arrecare alcun danno all'uomo, devono mantenere una distanza di sicurezza dall'uomo con una velocità adeguata. Questi valori sono dipendenti l'uno dall'altro e sono riportati nella normativa ISO/TS 15066.

Per replicare i movimenti dei due robot nell'ambiente di simulazione rispettando la suddetta normativa, è stato appositamente realizzato un algoritmo di controllo in cui il software Matlab esegue un apposito script e si interfaccia con Coppelasim grazie al processo di "chiamata API". Con questo metodo, è possibile delegare i calcoli e i passaggi logici allo script realizzato su Matlab, che restituisce in output i movimenti degli operatori. Tali informazioni vengono quindi inviate al software Coppelasim che le riproduce nell'ambiente di simulazione.

Lo script Matlab è costituito da un ciclo while che si ripete per un numero indefinito di volte, imponendo ai robot di raggiungere le posizioni e le pose richieste a monte, verificando costantemente se le posizioni e le pose richieste sono state raggiunte, monitorando la distanza tra uomo-robot mobile e uomo-robot antropomorfo per verificare il rispetto della normativa e correggendo le loro traiettorie se necessario. In questo modo, i robot possono muoversi secondo le richieste imposte da progetto e allo stesso tempo non urtare l'uomo pur condividendo lo stesso spazio di lavoro, indipendentemente dalle posizioni richieste ai due operatori.

Una caratteristica di questo algoritmo di controllo è il metodo con cui si riproducono i movimenti degli operatori nell'ambiente di simulazione. Questo metodo consiste nell'associare ad ogni ciclo dello script una posa e una posizione nello spazio (sia per il manichino che per i due robot), che vengono riprodotte sequenzialmente nell'ambiente di simulazione. Si formerà quindi una successione di frame che supera i 16 frame al secondo, soglia oltre cui lo spettatore ha l'impressione di un movimento continuo [21]. Si nota però che, nel ciclo while in questione, il movimento continuo avrà una velocità dipendente dalla

velocità di calcolo di Matlab, quindi slegata dalle velocità effettive. Si risolve questo problema imponendo un tempo di attesa tra un ciclo e il successivo. Aumentando questo intervallo di tempo dt i modelli nell'ambiente Coppeliassim si muovono più lentamente, fino a raggiungere una velocità adeguata.

Un altro punto degno di nota dell'algoritmo è l'acquisizione dei movimenti. Se per i due robot il risultato è il frutto di calcoli matematici, per il manichino l'output è di tipo sperimentale. Tramite l'uso di due telecamere Kinect, in laboratorio si è effettuata la registrazione dei movimenti di un operatore approssimandolo ad uno "skeleton" costituito da 25 punti [18]. L'acquisizione è stata effettuata da entrambe le Kinect, nello stesso intervallo di tempo e alla stessa frequenza di campionamento. L'output è costituito da due "skeleton", dai quali si otterrà uno "skeleton" ottimizzato, che a sua volta viene processato da un'apposita function realizzata su Matlab [16]].

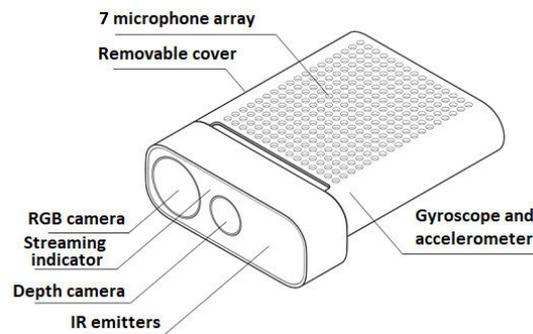


Figura 3.2: Telecamera Kinect

Il risultato finale è una successione di set di angoli riferiti ai giunti che vengono inviati all'ambiente Coppeliassim e permettono al manichino di muoversi.

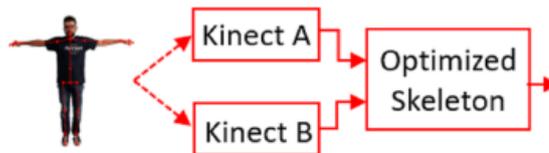


Figura 3.3: Schema per il calcolo del "Optimized skeleton"

3.1 Modello del TiaGo Robot

Per rappresentare il TiaGo in Coppeliassim, esso è stato approssimato come un cilindro, avendo il robot reale una forma molto simile a tale solido. Inoltre, sono state rappresentate le 2 ruote maggiori del robot, con i baricentri giacenti sulla stessa retta. Le dimensioni del cilindro, la sua quota rispetto al terreno e le dimensioni delle ruote sono state adottate

in modo da renderlo più somigliante possibile al robot reale, seguendo i dati presenti nel catalogo. Il sistema di riferimento del modello è solidale con il baricentro del suddetto cilindro, mentre la posa è stata orientata come segue:

- asse z: normale al piano, con verso diretto verso l'alto
- asse x: perpendicolare all'asse z e alla retta che unisce le due ruote più grandi, il verso è concorde con la direzione di avanzamento del TiaGo
- asse y: vincolato alla scelta dei primi due secondo la regola della mano destra.

Tale convenzione permette di ricavare il verso in cui il TiaGo è orientato rispetto al terreno. La posa del sistema di riferimento del robot rispetto al terreno è infatti calcolabile come:

$$A = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Dove θ è l'angolo di rotazione del sistema di riferimento intorno al proprio asse z. Per definizione, la prima colonna rappresenta il versore dell'asse x del sistema di riferimento del TiaGo nel sistema di riferimento O dell'ambiente di simulazione. Tale scelta consente di conoscere, noto θ in un dato istante di tempo, la direzione e il verso del TiaGo nel medesimo istante. Questa scelta tornerà utile nel calcolo delle velocità lineari e angolari da impartire ciclicamente al robot mobile.

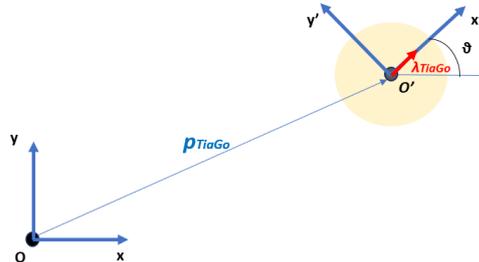


Figura 3.4: Posizione e orientamento del TiaGo nel piano

3.2 Campi Potenziali

Per ottenere un movimento da parte del TiaGo capace di raggiungere le coordinate spaziali desiderate ed evitare al contempo gli ostacoli lungo il tragitto sono stati utilizzati gli “Artificial Potential Field” (Campi Potenziali Artificiali). Un campo potenziale è una funzione dipendente dalle coordinate spaziali x e y che favorisce o ostacola il moto di un punto nello spazio. L’impatto del campo sul moto è dettato dal suo gradiente, che fornisce la velocità istantanea che il campo applicata al punto d’interesse in base alle sue coordinate spaziali.

I campi potenziali si dividono principalmente in:

- Attrattivi
- Repulsivi

Associando al punto di arrivo prescelto un campo attrattivo e agli ostacoli da evitare uno repulsivo si ottiene un campo totale che fungerà da “driving force” per il TiaGo.

Il punto di arrivo del TiaGo crea un campo attrattore del tipo:

$$U_{\text{base}}(x, y) = \frac{\sigma}{2} |p_{\text{TiaGo}} - p_{\text{arrivo}}|^2 \quad (3.1)$$

$$\nabla U_{\text{base}} = \sigma(p_{\text{TiaGo}} - p_{\text{arrivo}}) \quad (3.2)$$

$$|\nabla U_{\text{base}}| = \sigma |p_{\text{TiaGo}} - p_{\text{arrivo}}| \quad (3.3)$$

$$\lambda_{\text{base}} = -\frac{\nabla U_{\text{base}}}{|\nabla U_{\text{base}}|} \quad (3.4)$$

$$p_{\text{TiaGo}} = \begin{bmatrix} x_{\text{TiaGo}} \\ y_{\text{TiaGo}} \end{bmatrix}, \quad p_{\text{arrivo}} = \begin{bmatrix} x_{\text{arrivo}} \\ y_{\text{arrivo}} \end{bmatrix} \quad (3.5)$$

- p_{TiaGo} è il vettore posizione del TiaGo in una configurazione generica. La componente lungo l'asse x è pari a x_{TiaGo} , la componente lungo l'asse y è y_{TiaGo}
- p_{arrivo} è il vettore posizione del punto in cui il TiaGo deve arrivare. La componente lungo l'asse x è pari a x_{arrivo} , la componente lungo l'asse y è y_{arrivo}
- σ è un parametro scelto a monte che regola l'intensità del campo attrattivo e la velocità del TiaGo in funzione della distanza $|p_{\text{TiaGo}} - p_{\text{arrivo}}|$
- λ_{base} è il versore opposto al versore del gradiente di U_{base}

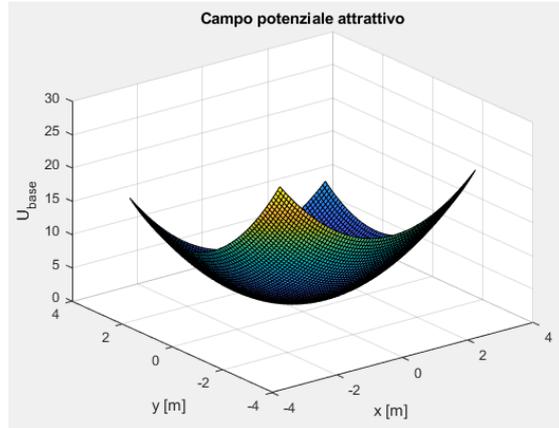


Figura 3.5: Campo potenziale attrattivo U_{base}

Il gradiente ∇U_{base} è un vettore costituito da:

- Un modulo $|\nabla U_{\text{base}}|$ proporzionale alla $|p_{\text{TiaGo}} - p_{\text{arrivo}}|$.

- Un vettore λ_{base} che parte dal baricentro del TiaGo e punta verso il punto di arrivo.

$$v_{\text{base}} = -\nabla U_{\text{base}} = |\nabla U_{\text{base}}| \lambda_{\text{base}} \quad (3.6)$$

Con questo metodo il TiaGo viene costantemente “spinto” verso la destinazione pre-impostata nell’ambiente 3D in Coppelasim e man mano che ci si avvicina, subisce un rallentamento progressivo. Tale metodo porta ad avere due vantaggi:

- Rende un’eventuale applicazione reale fattibile, permettendo al robot reale di poter rallentare progressivamente. Al contrario, se la velocità di avanzamento fosse costante, sarebbe necessaria un arresto istantaneo una volta raggiunte le coordinate desiderate
- Quando il robot è in movimento ma ad una velocità sufficientemente bassa, è possibile arrestare il moto ad una distanza ϵ trascurabile. Tale metodo è applicabile anche al robot reale, nel quale l’arresto brusco equivale all’azionamento dei freni del robot.

Per evitare la collisione del TiaGo con gli altri elementi presenti nell’ambiente di simulazione, sono stati adoperati dei campi potenziali repulsivi. Tali campi possono assumere diverse forme, ognuna delle quali con una formula diversa, anche se tutti sono accomunati da:

- Un coefficiente K che indica l’intensità del campo
- Un coefficiente γ che indica il decadimento del campo potenziale

Per il momento non ci si sofferma sui valori che questi coefficienti devono avere. Il tema sarà trattato più avanti, quando verrà illustrato l’algoritmo per la *collision avoidance* nella sua interezza.

Per l’applicazione in questione i due ostacoli tenuti in considerazione, ovvero il manichino e il tavolo, sono stati assimilati rispettivamente ad un cilindro [15] e ad una sigmoide [19]. Di queste due figure geometriche è possibile definire un campo potenziale repulsivo come riportato sotto:

$$U_{\text{manichino}}(x, y) = K_M e^{-\frac{\gamma_M}{2}((x_{\text{TiaGo}} - x_{\text{Human}})^2 + (y_{\text{TiaGo}} - y_{\text{Human}})^2)} \quad (3.7)$$

Cilindro:

$$\nabla U_{\text{manichino}} = -\gamma_M K_M e^{-\frac{\gamma_M}{2}((x_{\text{TiaGo}} - x_{\text{Human}})^2 + (y_{\text{TiaGo}} - y_{\text{Human}})^2)} \begin{bmatrix} x_{\text{TiaGo}} - x_{\text{Human}} \\ y_{\text{TiaGo}} - y_{\text{Human}} \end{bmatrix} \quad (3.8)$$

$$U_{\text{tavolo}}(x, y) = K_T \prod_{i=1}^4 \frac{1}{1 + e^{-\gamma_T(A_i(x_{\text{TiaGo}} - x_{\text{Tavolo}}) + B_i(y_{\text{TiaGo}} - y_{\text{Tavolo}}) + C_i)}} \quad (3.9)$$

Sigmoide:

$$\nabla U_{\text{tavolo}} = -\gamma_T \cdot U_{\text{tavolo}} \sum_{i=1}^4 (1 - U_i) \begin{bmatrix} A_i \\ B_i \end{bmatrix} \quad (3.10)$$

In cui:

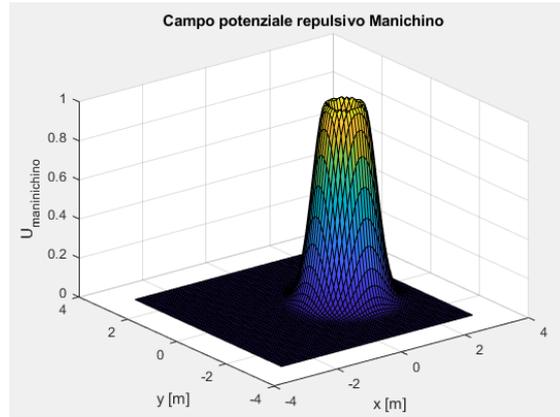


Figura 3.6: Campo potenziale repulsivo del manichino

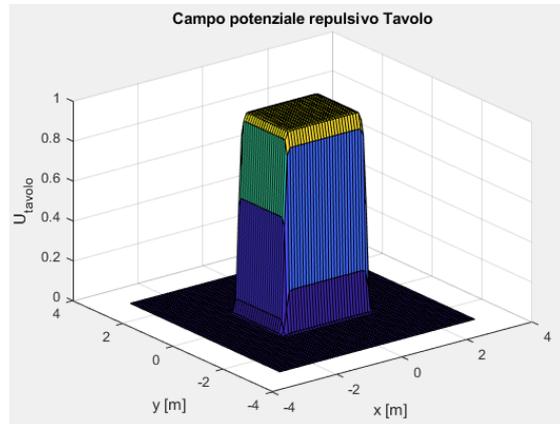


Figura 3.7: Campo potenziale repulsivo del tavolo

- K_M e γ_M sono rispettivamente i coefficienti di intensità e di decadimento del campo potenziale riferito al manichino; K_T e γ_T sono rispettivamente i coefficienti di intensità e di decadimento del campo potenziale riferito al tavolo
- x_{Tavolo} e y_{Tavolo} sono le coordinate del baricentro del tavolo
- x_{Human} e y_{Human} sono le coordinate del baricentro del manichino
- A_i , B_i , e C_i sono coefficienti che dipendono dalle dimensioni del tavolo [19]

Per il principio di sovrapposizione degli effetti, si possono sommare i tre campi potenziali:

$$U_{tot} = U_{base} + U_{manichino} + U_{tavolo} \quad (3.11)$$

Calcolando il gradiente del potenziale U_{tot} si ricava la velocità effettiva del TiaGo:

$$v_{Tot} = -\nabla U_{Tot} = -\nabla U_{base} - \nabla(U_{manichino} + U_{tavolo}) \quad (3.12)$$

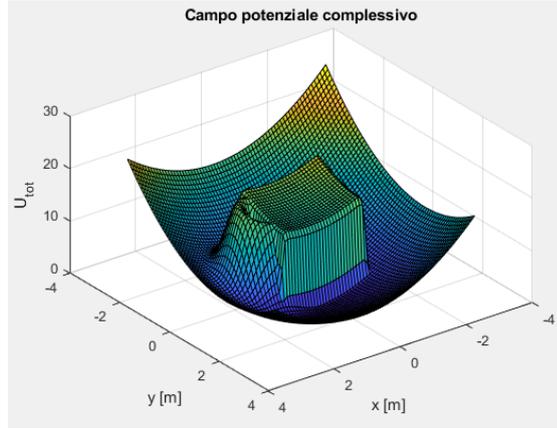


Figura 3.8: Campo potenziale complessivo

Essendo il potenziale la “driving force” dell’intero moto, da esso si ricava anche il percorso che il robot dovrà seguire per arrivare a destinazione. Applicando il gradiente del campo potenziale U_{Tot} si ottiene anche il modulo della velocità istantanea effettiva richiesta al TiaGo per un qualsiasi punto nello spazio.

Si può anche ricavare l’influenza che i campi repulsivi hanno in termini di “velocità repulsiva”:

$$v_{rep} = v_{Tot} - v_{base} = -\nabla(U_{manichino} + U_{tavolo}) \quad (3.13)$$

Alla velocità repulsiva sono associabili anche i versori:

$$\lambda_{rep} = -\frac{v_{rep}}{|v_{rep}|} \quad (3.14)$$

$$\lambda_{coll} = -\lambda_{rep} \quad (3.15)$$

Tale metodo, tuttavia, può risultare inadeguato per l’applicazione richiesta, principalmente per due ragioni:

1. Efficienza: il robot viene guidato dal solo potenziale, ma così può capitare che non venga scelto il percorso più breve, allungando quindi i tempi dell’intera operazione
2. Minimi locali: la somma di più campi potenziali privi di minimi locali può portare a un campo potenziale con uno o più minimi locali [14]

Se il primo punto non comporta problemi di tipo realizzativo, intaccando solo la ricerca della soluzione ottimale, lo stesso non si può dire per il secondo. Il raggiungimento di un minimo locale priva il robot mobile della sua “driving force”, impedendogli di raggiungere la posizione richiesta. Ciò avviene in quanto la velocità del TiaGo è pari al gradiente del potenziale U_{tot} .

$$U_{Tot}(x^*, y^*) = \text{cost} \quad (3.16)$$

$$-\nabla U_{Tot}(x^*, y^*) = 0 \quad (3.17)$$

$$-\nabla U_{\text{base}} - \nabla(U_{\text{manichino}} + U_{\text{tavolo}}) = 0 \quad (3.18)$$

$$\mathbf{v}_{\text{base}} + \mathbf{v}_{\text{rep}} = 0 \quad (3.19)$$

$$\mathbf{v}_{\text{rep}} = -\mathbf{v}_{\text{base}} \quad (3.20)$$

Se quindi in un punto nel piano U_{Tot} ha un punto di minimo, in quello stesso punto \mathbf{v}_{rep}

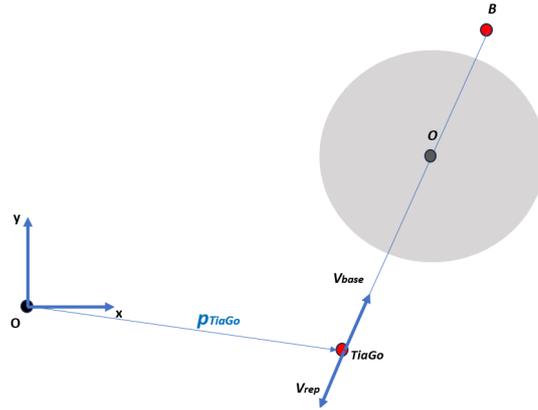


Figura 3.9: Esempio di situazione in cui $(v_{\text{Tot}})=0$

è pari a \mathbf{v}_{base} con segno opposto, impedendogli di raggiungere la posizione richiesta.

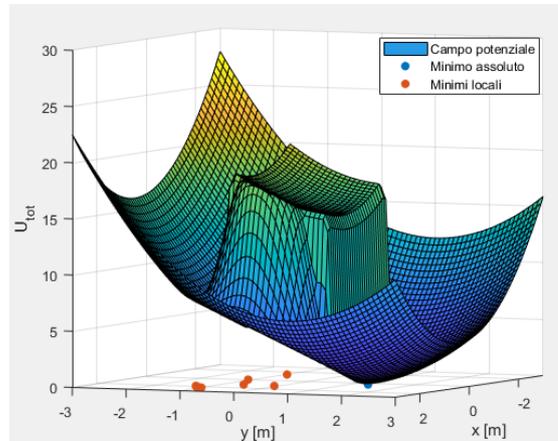


Figura 3.10: Campo potenziale complessivo

3.3 Applicazione della Collision Avoidance al TiaGo

Dinanzi a questa criticità è quindi necessario riadattare l'uso dei campi attrattivi e repulsivi con un nuovo metodo che elimini il problema dei minimi locali, oltre a ricavare il percorso ottimale, slegando quindi il problema dal semplice calcolo del gradiente di un

potenziale. A tale scopo, da questo momento in poi e fino a nuove indicazioni, il TiaGo robot è ridotto a un punto materiale libero di muoversi nell'ambiente di lavoro e di cambiare anche istantaneamente il proprio orientamento. Inoltre, si ipotizza di avere a che fare con un ostacolo immobile.

Per risolvere il problema del percorso ottimale da assegnare al TiaGo, si ipotizza la presenza di un punto A di partenza, di un punto B di arrivo e di un ostacolo di forma circolare di centro O', si avrà una situazione analoga a quella rappresentata in 3.11. In questa

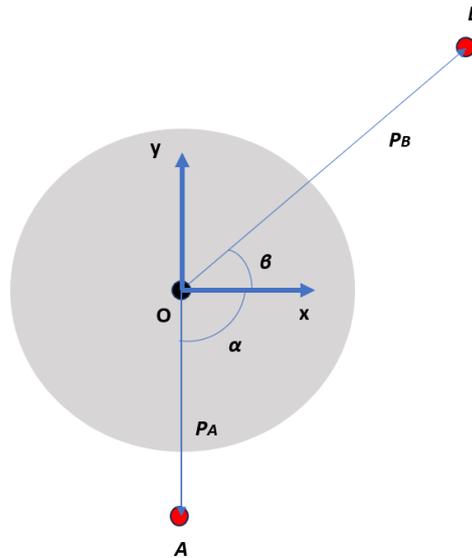


Figura 3.11: Configurazione generica dei punti A e B con ostacolo di forma circolare

configurazione:

- \mathbf{p}_A e \mathbf{p}_B sono i vettori posizione relativi ad A e B
- α e β sono gli angoli di rotazione di \mathbf{p}_A e \mathbf{p}_B rispetto alla normale al piano

Si nota che, affinché la direzione del vettore \mathbf{p}_A coincida con quella di \mathbf{p}_B , è possibile far ruotare il vettore attorno ad O in due modi diversi

I punti A e B sono rispettivamente i punti di partenza e di arrivo di P in una configurazione generica, identificati dai vettori posizione \mathbf{v}_A e \mathbf{v}_B e dagli angoli alfa e beta. Si nota che, affinché la direzione del vettore \mathbf{v}_A coincida con quella di \mathbf{v}_B , è possibile far ruotare il vettore attorno ad O in due modi diversi.

- In senso orario di un angolo γ_1 pari alla somma dei valori assoluti di α e β .
- In senso antiorario di un angolo γ_2 supplementare a γ_1 . Trascurando il caso limite con γ_1 e γ_2 pari a 180° , in cui non vi è una direzione privilegiata, esisterà sempre un verso di rotazione che, a parità di velocità, permetterà a un punto materiale in A di raggiungere il punto B più rapidamente rispetto all'altro.

Calcolando la somma dei valori assoluti di α e β , si valuta se:

- la somma dei loro valori assoluti è maggiore o minore di 180°
- Se il vettore \mathbf{p}_A deve ruotare in senso orario o antiorario affinché la sua nuova direzione coincida con quella del vettore \mathbf{p}_B .

Queste due condizioni portano a quattro possibili situazioni, illustrate in 3.12, 3.13, 3.14 e 3.15, in cui è riportata, caso per caso, come il robot aggira l'ostacolo.

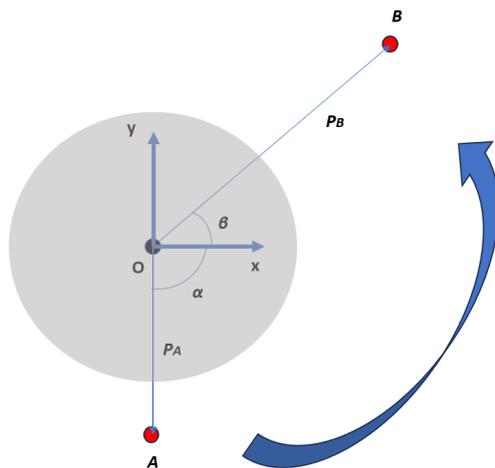


Figura 3.12: Scelta percorso ottimale (caso 1)

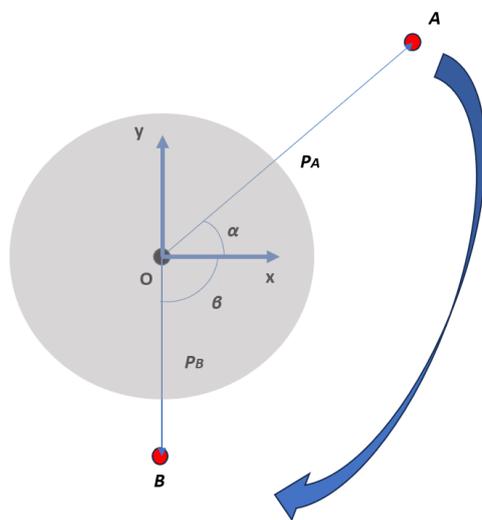


Figura 3.13: Scelta percorso ottimale (caso 2)

Il risultato è un percorso preferenziale che non cambierà durante l'intero percorso, essendo

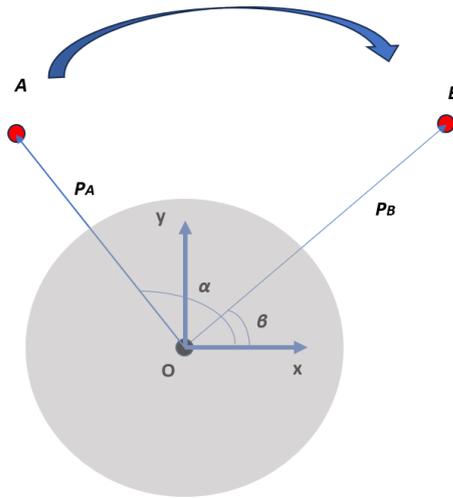


Figura 3.14: Scelta percorso ottimale (caso 3)

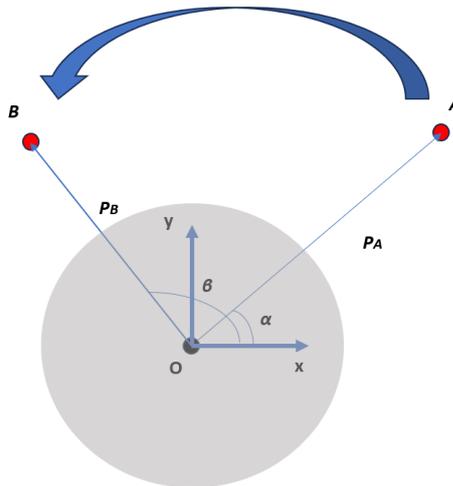


Figura 3.15: Scelta percorso ottimale (caso 4)

dipendente dalle sole coordinate dei punti A e B.

Nel caso di studio d'interesse, essendo l'ostacolo da aggirare un tavolo, invece di un ostacolo di forma circolare è presente un ostacolo di forma rettangolare, ma il principio di base è lo stesso.

Risolto il problema dell'ottimizzazione del percorso, rimane da risolvere il problema di maggior rilievo per l'applicazione studiata, ovvero la collision avoidance per evitare eventuali ostacoli lungo il percorso. Occorre inoltre che il nuovo metodo si adatti a quanto appena spiegato.

Ipotizzando che il TiaGo si muova a velocità costante v_{base} , esso continuerà ad avanzare

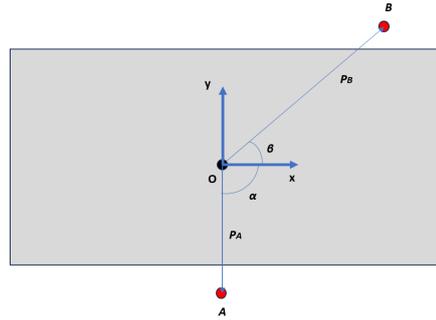


Figura 3.16: Configurazione generica dei punti A e B con ostacolo di forma rettangolare

nella stessa direzione e la sua distanza sarà direttamente proporzionale al tempo impiegato. In presenza di un ostacolo, però, a una data distanza, entrerà in gioco una velocità repulsiva \mathbf{v}_{rep} . Tale velocità è applicata al TiaGo e, sommandosi alla velocità di base, restituisce un nuovo valore di velocità, che può variare sia in modulo che in verso rispetto alla velocità di partenza.

$$\mathbf{v}_{\text{TOT}} = \mathbf{v}_{\text{base}} + \mathbf{v}_{\text{rep}} \quad (3.21)$$

È però evidente che ricavare la \mathbf{v}_{TOT} in questo modo non è una soluzione accettabile.

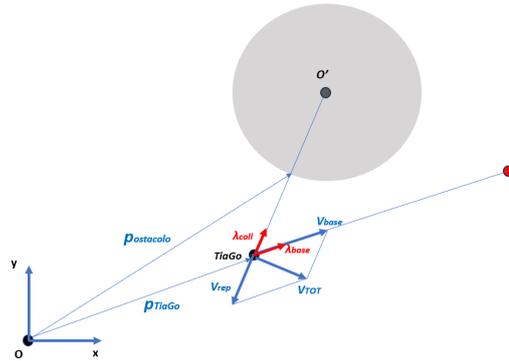


Figura 3.17: Configurazione in cui $v_{\text{Tot}} = v_{\text{base}} + v_{\text{rep}}$

Così, infatti si avrà che:

$$|\mathbf{v}_{\text{Tot}}| \neq |\mathbf{v}_{\text{base}}| \quad (3.22)$$

In questo modo il modulo della velocità imposto al TiaGo cambia in modo incontrollato. Ne consegue che solo λ_{Tot} risente dell'effetto repulsivo e cambia direzione rispetto a λ_{base} , mentre $|\mathbf{v}_{\text{Tot}}|$ dipenderà esclusivamente dalla posizione nello spazio.

Inoltre, si può notare che la distanza dall'ostacolo non è l'unica condizione che determina l'intervento della velocità \mathbf{v}_{rep} sul TiaGo.

Nell'ipotesi di una \mathbf{v}_{base} orientata come rappresentato in 3.19 la distanza dall'ostacolo non

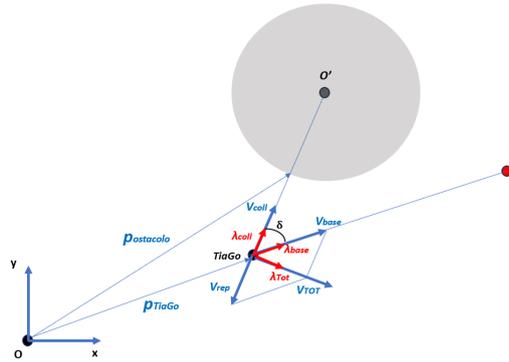


Figura 3.18: Configurazione in cui $|v_{Tot}| = |v_{base}|$ indipendentemente da v_{rep}

comporta rischi di urto con l'ostacolo in quanto il versore λ_{base} porta il robot ad allontanarsi sempre di più dall'ostacolo. Per questi motivi, nello studio delle velocità repulsive, è

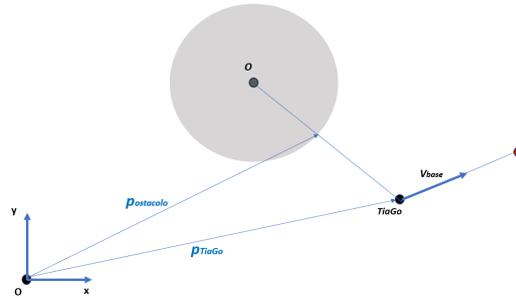


Figura 3.19: Caso in cui non è necessario l'intervento di collision avoidance

sufficiente tenere in considerazione la sola componente della \mathbf{v}_{base} orientata lungo la direzione in cui avrà effetto la velocità repulsiva. Tale velocità è definita \mathbf{v}_{coll} ed è calcolabile come:

$$\mathbf{v}_{coll} = |\mathbf{v}_{base}|(\boldsymbol{\lambda}_{base} \cdot \boldsymbol{\lambda}_{coll})\boldsymbol{\lambda}_{coll} \quad (3.23)$$

Il prodotto scalare in questione può essere maggiore, pari o minore di zero. Se negli ultimi due casi l'ostacolo è trascurabile per quanto spiegato sopra, nel primo caso invece contribuisce al calcolo di \mathbf{v}_{rep} .

Noto $|\mathbf{v}_{coll}|$, è possibile calcolare la velocità \mathbf{v}_{rep} con un altro metodo, che si serve di un valore che, in questo lavoro, è stato denominato "Indice di repulsione". Per "Indice di repulsione", si intende un numero reale, chiamato r , che è definito come:

$$r = \frac{|\mathbf{v}_{rep}|}{|\mathbf{v}_{coll}|} \quad (3.24)$$

Con r è possibile definire la prossimità di un punto materiale in movimento a un qualunque ostacolo. Per definizione, l'indice di repulsione può variare solo tra 0 e 1 estremi

compresi, dove 0 corrisponde ad una distanza teoricamente infinita dall'ostacolo, mentre 1 corrisponde alla distanza minima accettabile. Con esso è possibile servirsene per calcolare λ_{tot} servendosi dei campi potenziali e potendo scegliere il percorso ottimale per l'aggiramento dell'ostacolo, senza però cadere nel problema dei minimi locali.

Questo metodo risulta efficace in quanto permette di bypassare il problema della definizione dei coefficienti K dei campi potenziali repulsivi spiegati in precedenza. Nota infatti la distanza $|\mathbf{p}_{\text{TiaGo}} - \mathbf{p}_{\text{ostacolo}}|$ e il versore $\boldsymbol{\lambda}_{\text{base}}$, anche imponendo una distanza minima accettabile, non è possibile calcolare la velocità \mathbf{v}_{rep} senza conoscere K , ma è possibile calcolare r come:

$$r = \frac{|\mathbf{v}_{\text{rep}}|}{|\mathbf{v}_{\text{rep limite}}|} \quad (3.25)$$

In questa situazione, il fattore K è infatti ininfluenza. Si noti che $|\mathbf{v}_{\text{rep limite}}|$ è la velocità repulsiva che il campo repulsivo esercita sul TiaGo tale da impedire che $|\mathbf{p}_{\text{TiaGo}} - \mathbf{p}_{\text{ostacolo}}|$ diminuisca ulteriormente. Per definizione, quindi, $|\mathbf{v}_{\text{rep limite}}|$ è pari $|\mathbf{v}_{\text{coll}}|$ al netto di una costante. A questo punto si può calcolare \mathbf{v}_{Tot} come:

$$\mathbf{v}_{\text{interm}} = \mathbf{v}_{\text{base}} + |\mathbf{v}_{\text{rep}}|\boldsymbol{\lambda}_{\text{rep}} \quad (3.26)$$

$$\mathbf{v}_{\text{interm}} = \mathbf{v}_{\text{base}} + r|\mathbf{v}_{\text{coll}}|\boldsymbol{\lambda}_{\text{rep}} = \mathbf{v}_{\text{base}} - r|\mathbf{v}_{\text{coll}}|\boldsymbol{\lambda}_{\text{coll}} \quad (3.27)$$

$$\mathbf{v}_{\text{Tot}} = \frac{|\mathbf{v}_{\text{base}}|\mathbf{v}_{\text{interm}}}{|\mathbf{v}_{\text{interm}}|} \quad (3.28)$$

Tale procedimento, però, continua a risentire del problema dei minimi locali, arrestando il TiaGo in determinati punti, in cui:

$$\lambda_{\text{base}} = \lambda_{\text{coll}} \quad (3.29)$$

Esso porta alla seguente caso limite:

$$\mathbf{v}_{\text{base}} - r|\mathbf{v}_{\text{coll}}|\boldsymbol{\lambda}_{\text{coll}} = (|\mathbf{v}_{\text{base}}| - r|\mathbf{v}_{\text{coll}}|)\boldsymbol{\lambda}_{\text{coll}} = \mathbf{v}_{\text{Tot}} = |\mathbf{v}_{\text{Tot}}|\boldsymbol{\lambda}_{\text{coll}} \quad (3.30)$$

$$|\mathbf{v}_{\text{base}}| - r|\mathbf{v}_{\text{coll}}| = |\mathbf{v}_{\text{Tot}}| = 0 \quad (3.31)$$

$$r = \frac{|\mathbf{v}_{\text{base}}|}{|\mathbf{v}_{\text{coll}}|} \quad (3.32)$$

Essendo $|\mathbf{v}_{\text{coll}}| \leq |\mathbf{v}_{\text{base}}|$ e $r \geq 1$ per definizione, il risultato è che i minimi locali si verificano in situazioni in cui:

$$r = 1 \quad (3.33)$$

Occorre quindi trovare un nuovo metodo per aggirare questo caso limite, cercando di sfruttare l'indice di repulsione r in un altro modo. Questo nuovo metodo consiste nel definire, per ogni versore un λ_{coll} , un altro versore λ_{tan} tale che:

$$\boldsymbol{\lambda}_{\text{coll}} \cdot \boldsymbol{\lambda}_{\text{tan}} = 0 \quad (3.34)$$

Se come già detto, con r pari a 1 il TiaGo non può avvicinarsi ulteriormente all'ostacolo, e con r nullo l'ostacolo non influisce sulla velocità complessiva, ne consegue che:

$$\mathbf{v}_{\text{Tot}}(r = 0) = |\mathbf{v}_{\text{base}}|\boldsymbol{\lambda}_{\text{base}} \quad (3.35)$$

$$\mathbf{v}_{\text{Tot}}(r = 1) = |\mathbf{v}_{\text{base}}| \boldsymbol{\lambda}_{\text{tan}} \quad (3.36)$$

È quindi possibile legare il versore λ a r come:

$$u(r) = \lambda_{\text{tan}} r + (1 - r) \lambda_{\text{base}} \quad (3.37)$$

$$\lambda(r) = \frac{u(r)}{|u(r)|} \quad (3.38)$$

Rimane quindi da definire λ_{tan} , che può essere identificato da due versori opposti tra loro ed entrambi perpendicolari nel piano a λ_{coll} . La scelta è effettuata scegliendo quello che ottimizza il percorso del TiaGo ed è calcolabile come segue:

$$\lambda_{\text{tan}} = M \cdot \lambda_{\text{coll}} \quad (3.39)$$

$$M = \begin{cases} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, & \text{if } |\alpha| + |\beta| \leq 180^\circ \text{ and } \alpha \leq \beta \\ \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, & \text{if } |\alpha| + |\beta| \leq 180^\circ \text{ and } \alpha > \beta \end{cases} \quad (3.40)$$

Questo metodo, tuttavia, obbliga il TiaGo ad aggirare l'ostacolo permanentemente, essendo sprovvisto di un fattore che faccia allontanare il robot dall'ostacolo oltre che rallentare l'avvicinamento.

La soluzione adottata prevede di impiegare entrambi i metodi descritti che prevedono l'uso dell'indice r : inizialmente è favorito il metodo di aggiramento dell'ostacolo. In seguito, man mano che il TiaGo aggira l'ostacolo, si avvicinerà al punto B di arrivo, fino a raggiungere una distanza tale per cui non esiste il rischio di ricadere in alcun minimo locale; quindi, è possibile applicare il metodo classico delle velocità repulsive.

Finora si è assimilato il TiaGo a un punto materiale libero di muoversi su un piano per studiare come muoverlo da un punto A ad un punto B evitando eventuali ostacoli lungo il cammino. Tale ipotesi semplificativa però non può essere mantenuta ulteriormente, in quanto un robot mobile è caratterizzato da una direzione di avanzamento ben precisa, che è definita come λ_{TiaGo} , diversa dalla λ_{tot} calcolata finora. Nel dettaglio:

- λ_{tot} è il versore che indica l'orientamento richiesto al TiaGo
- λ_{TiaGo} è il versore che indica l'orientamento effettivo del TiaGo

L'angolo compreso tra λ_{tot} è denominato ϕ ed è definito come:

$$\phi = \arccos(\lambda_{\text{tot}} \cdot \lambda_{\text{TiaGo}}) \quad (3.41)$$

In una configurazione generica, ϕ è non nullo, quindi il TiaGo è orientato diversamente da quanto richiesto dai campi potenziali. Ne consegue che il TiaGo dovrà anche orientare λ_{TiaGo} fino alla situazione:

$$\lambda_{\text{tot}} \cdot \lambda_{\text{TiaGo}} = 1 \quad (3.42)$$

Ciò, tuttavia fa emergere una serie di complicazioni:

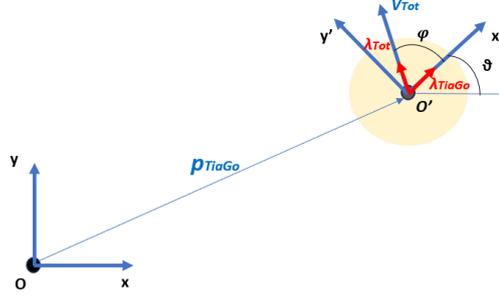


Figura 3.20: Schema del TiaGo con il versore imposto dai campi potenziali e il versore effettivo del robot

- Il robot deve essere in grado di ruotare intorno al proprio asse z, normale al piano con verso uscente dallo stesso, con velocità angolare appropriata e non infinita
- Una rotazione di velocità finita crea una serie di configurazioni intermedie tra l'orientamento di partenza del robot e l'orientamento desiderato.
- Per riprodurre nella simulazione un movimento realistico da parte del robot, serve stabilire istante per istante come variare la velocità lineare e angolare

Per risolvere questi problemi, sono state ideate le seguenti formule per il controllo della velocità lineare e angolare:

$$\omega = C_\omega \omega_{\max} \quad (3.43)$$

$$v = C_v v_{\max} \quad (3.44)$$

$$C_\omega = \begin{cases} K\phi, & \phi < 1 \\ 1, & \phi \geq 1 \end{cases} \quad (3.45)$$

$$C_v = \begin{cases} 1 - \phi, & \phi \geq 1 \\ 0, & \phi < 1 \end{cases} \quad (3.46)$$

La legge di controllo consiste nel moltiplicare ω e v per dei coefficienti dipendenti da ϕ . Il coefficiente C_ω è direttamente proporzionale a ϕ , non andando oltre il valore 1. Al contrario, C_v diminuisce all'aumentare di ϕ , saturando su zero. La scelta di queste due leggi di controllo è stata validata da diverse simulazioni, che hanno riprodotto un movimento del TiaGo capace di bilanciare reattività e fluidità di movimento. La velocità angolare ω_{\max} è nota a monte, mentre v_{\max} è pari a:

$$v_{\max} = \begin{cases} v_{\text{limite}}, & |v_{\text{tot}}| \geq v_{\text{limite}} \\ |v_{\text{tot}}|, & |v_{\text{tot}}| < v_{\text{limite}} \end{cases} \quad (3.47)$$

in cui v_{limite} è la velocità limite ammessa dal TiaGo reale, ovvero 1 m/s.

Da questo si deduce che la dipendenza tra v e r è implicita, ovvero che:

$$v(r) = v(r(\phi)) = v(\phi) \quad (3.48)$$

Ciò significa che, se ϕ è nullo, la velocità del TiaGo, pur cambiando direzione, non cambia il suo modulo anche se si sta avvicinando ad un ostacolo. Si è provato che, in tale circostanza, l'algoritmo sopra descritto cade in errore. Per risolvere questo problema si è calcolato un secondo "Indice di repulsione", chiamato r_{TiaGo} . I passaggi per il calcolo di r_{TiaGo} sono analoghi a quelli di r , con la differenza che, nei vari passaggi, λ_{base} è sostituito da λ_{TiaGo} .

$$v_{\text{coll TiaGo}} = |v_{\text{base}}|(\lambda_{\text{TiaGo}} \cdot \lambda_{\text{coll}})\lambda_{\text{coll}}$$

Calcolato r_{TiaGo} , si può calcolare il modulo della velocità di avanzamento del TiaGo effettiva, denominata v_{TiaGo} come segue:

$$|v_{\text{TiaGo}}| = \begin{cases} v - |v_{\text{coll TiaGo}}|r_{\text{TiaGo}}, & v \geq |v_{\text{coll TiaGo}}|r_{\text{TiaGo}} \\ 0, & v < |v_{\text{coll TiaGo}}|r_{\text{TiaGo}} \end{cases}$$

Per quanto riguarda invece la velocità angolare del TiaGo si ha che:

$$\omega_{\text{TiaGo}} = \omega$$

3.4 Applicazione della Normativa al TiaGo

Finora ci si è limitati, a studiare casi in cui il TiaGo deve raggiungere un punto nello spazio, seguendo il percorso più breve e dovendo evitare ostacoli fermi nello spazio. Se però l'obiettivo della collision avoidance è permettere al robot di lavorare senza causare problemi all'operatore umano con cui condivide la zona di lavoro, bisogna supporre che la persona deve essere libera di muoversi liberamente nello spazio, senza vincoli di alcuna natura. Dal punto di vista del robot, ci si ritrova davanti un ostacolo mobile, dal quale deve rimanere lontano di una distanza minima che è funzione della propria velocità. In virtù dell'impossibilità da parte del TiaGo di urtare la parte superiore della persona, si è scelto di assimilare quest'ultima ad un ostacolo di forma cilindrica di raggio opportuno. In fig.55 è riportato il confronto tra il manichino e l'ostacolo di forma cilindrica al quale viene assimilato in ambiente Coppeliasim.

Questa semplificazione permette di trascurare la complessità del manichino, permette di poter usare il campo repulsivo di forma circolare $U_{\text{manichino}}$. Come indicato nella normativa ISO/TS 15066, esiste un legame tra la distanza uomo - robot e le velocità in gioco, e sintetizzata nella seguente formula:

$$d_{\text{min}} = (v_{\text{human}} + v_{\text{robot}}) \cdot T_r + \frac{v_{\text{robot}}^2}{2a_s} \quad (3.49)$$

Questa formula indica la distanza di sicurezza che il robot deve mantenere dalla persona. Nel caso in questione si è scelta la velocità del robot come variabile indipendente, mentre gli altri valori sono stati fissati. Essi sono:

- v_{human} è la velocità della persona
- T_r è il tempo di reazione del robot



Figura 3.21: Manichino e cilindro per l'applicazione della normativa

- a_s è il tempo di reazione del robot

Gli ultimi due valori possono essere facilmente ipotizzati costanti e sono stati scelti rispettivamente pari a 0,21 s e 10 m/s^2 , basandosi sullo studio di [3]. Lo stesso però non si può dire per la velocità v_{human} . Applicando rigorosamente la normativa, si è scelto il valore di riferimento di 1,6 m/s. Combinando la formula ottenuta dalla normativa con il campo repulsivo $U_{\text{manichino}}$, si può ricavare la velocità repulsiva applicando il concetto di indice di repulsione spiegato prima. I risultati intermedi sono i seguenti:

$$v_{\text{rep human}} = \gamma_M K_M d e^{-\frac{\gamma_M}{2} d^2} \quad \text{and} \quad (3.50)$$

$$v_{\text{rep human max}} = \gamma_M K_M d_{\min} e^{-\frac{\gamma_M}{2} d_{\min}^2} \quad (3.51)$$

$$d = \sqrt{(x_{\text{TiaGo}} - x_{\text{Human}})^2 + (y_{\text{TiaGo}} - y_{\text{Human}})^2} \quad (3.52)$$

Il risultato finale è un indice di repulsione calcolabile come segue:

$$r(d) = \frac{d}{d_{\min}} e^{-\frac{\gamma_M}{2} (d^2 - d_{\min}^2)} \quad (3.53)$$

Noto l'indice di repulsione, ci si può ricondurre ai passaggi già affrontati in precedenza per il caso di ostacolo non in movimento.

Per il calcolo del coefficiente di decadimento γ_M non si è applicato un metodo analitico: come nel caso della sigmoide, infatti, esso non impatta sul rispetto della normativa, ma solo la gradualità del processo. In questo caso, γ_M è stato scelto sperimentalmente come:

$$\gamma_M = \frac{1}{d_{\min}^2} \quad (3.54)$$

3.5 Modello del Robot Franka Emika FR3

Per riprodurre il robot antropomorfo nell'ambiente di simulazione 3D, si è attinto al modello presente nella libreria di Coppeliiasim, nella sezione "Robot fissi".

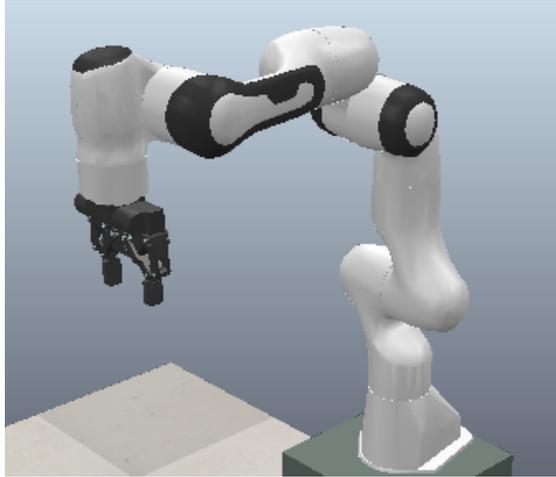


Figura 3.22: Robot FR3 in ambiente Coppeliiasim

Questo modello è costituito, come il robot reale, da 8 link collegati tra loro da 7 giunti rotoidali.

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \end{bmatrix}^T \quad (3.55)$$

Per la definizione dei sistemi di riferimento dei vari giunti ci si è basati sulla convenzione di Denavit-Hartenberg modificata, che ha portato come risultati i parametri di Denavit-Hartenberg in 3.23.

Qui si presenta dunque una prima semplificazione: essendo sufficienti 6 giunti per ottenere una qualsiasi posa tra quelle disponibili nel volume di lavoro, si è dinanzi a un problema di ridondanza, ovvero infinite soluzioni al problema di cinematica. Attingendo a un lavoro [6], si è scelto di bloccare il giunto a valle della catena (il giunto 7) fissando il suo valore a:

$$q_7 = 0 \quad (3.56)$$

Notare come questa semplificazione non impedisce al giunto q_7 di non ritagliarsi uno spazio in alcune applicazioni specifiche: se infatti al robot venisse richiesto di rimuovere un elemento di serraggio (es. una vite), esso non dovrebbe necessariamente servirsi di tutti i giunti per portare a termine il task. Per come è strutturato, infatti, il giunto più

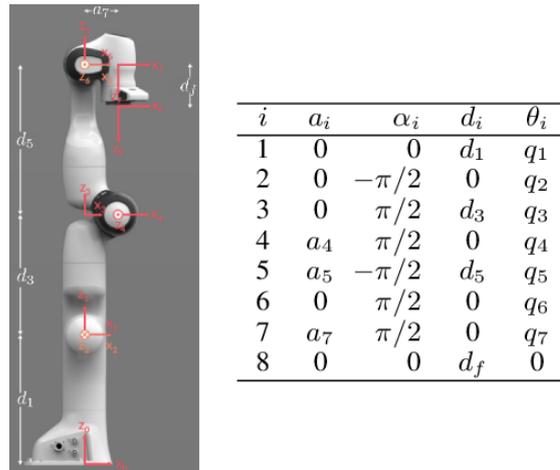
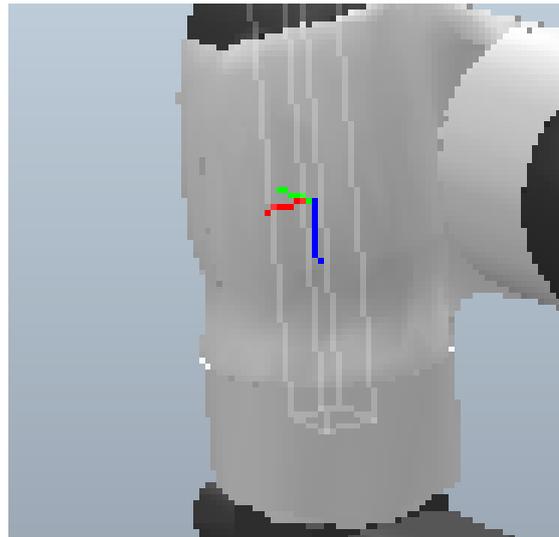


Figura 3.23: Robot F53 con i parametri di Denavit-Hartenberg

vicino all'end effector del robot presenta un asse z orientato come in 3.24.

Figura 3.24: Posa del sistema di riferimento associato al giunto q_7 , fissa rispetto al sistema di riferimento associato al giunto q_6

Questa struttura permette al robot di suddividere un task di rimozione vite in due parti legate dal punto di vista analitico, ma che lavorano all'unisono:

- La catena cinematica che va dal giunto 1 al giunto 6 fornisce la componente di traslazione spostando l'end effector verso l'alto
- Il giunto 7 fornisce la componente di rotazione, la cui intensità dipende da quella di traslazione, affinché lo svitamento venga eseguito correttamente

Con questa semplificazione è possibile applicare l'algoritmo di cinematica inversa studiato da [6].

Si noti che in questo lavoro, per il calcolo della cinematica diretta ci si riferisce al calcolo della cinematica diretta, della cinematica inversa e del calcolo della Jacobiana relativa al robot FR3 con le seguenti espressioni:

$$\hat{A} = \text{Forward_kinematic}(\mathbf{q}) \quad (3.57)$$

$$\mathbf{q} = \text{Inverse_kinematic}(\hat{A}) \quad (3.58)$$

$$J = \text{Jacobiana}(\mathbf{q}, \mathbf{p}) \quad (3.59)$$

In cui:

- \hat{A} è la matrice di posa 4 x 4, calcolata moltiplicando opportunamente tra loro le matrici:
- J è la Jacobiana di dimensioni 6 x 6, quindi trascurando il giunto q_7 . Per il calcolo della Jacobiana si fa riferimento a [29]
- \mathbf{p} è il vettore posizione utilizzato di cui si vuole calcolare la Jacobiana

$${}^0\mathbf{A}_1(q_1) = \begin{pmatrix} \cos q_1 & 0 & \sin q_1 & 0 \\ \sin q_1 & 0 & -\cos q_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad {}^1\mathbf{A}_2(q_2) = \begin{pmatrix} \cos q_2 & 0 & -\sin q_2 & 0 \\ \sin q_2 & 0 & \cos q_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$${}^2\mathbf{A}_3(q_3) = \begin{pmatrix} \cos q_3 & 0 & \sin q_3 & a_3 \cos q_3 \\ \sin q_3 & 0 & -\cos q_3 & a_3 \sin q_3 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad {}^3\mathbf{A}_4(q_4) = \begin{pmatrix} \cos q_4 & 0 & -\sin q_4 & a_4 \cos q_4 \\ \sin q_4 & 0 & \cos q_4 & a_4 \sin q_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

Figura 3.25: Matrici di posa del robot FR3 (parte 1)

3.6 Percorso dell'End Effector senza Collision Avoidance

Una volta che il robot mobile ha raggiunto il punto nel piano indicato dall'apposito puntatore posto nell'ambiente di simulazione, l'algoritmo si adopera per realizzare anche il percorso del robot FR3, secondo le specifiche imposte a monte. Come per il TiaGo, anche robot FR3 è il risultato di una successione di pose calcolate ad intervalli di tempo dt , con dt appositamente scelto per far muovere il modello in Coppeliassim a velocità coerenti con quelle reali, come spiegato ad inizio Capitolo.

Come ipotesi semplificativa, si è scelto di muovere l'end effector secondo un percorso rettilineo, partendo da una posizione p_{start} e terminando in una posizione p_{finish} . La posa iniziale A_{start} e quella finale A_{finish} sono le seguenti:

$${}^4\mathbf{A}_5(q_5) = \begin{pmatrix} \cos q_5 & 0 & \sin q_5 & 0 \\ \sin q_5 & 0 & -\cos q_5 & 0 \\ 0 & 1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad {}^5\mathbf{A}_6(q_6) = \begin{pmatrix} \cos q_6 & 0 & \sin q_6 & a_6 \cos q_6 \\ \sin q_6 & 0 & -\cos q_6 & a_6 \sin q_6 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$${}^6\mathbf{A}_7(q_7) = \begin{pmatrix} \cos q_7 & -\sin q_7 & 0 & 0 \\ \sin q_7 & \cos q_7 & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Figura 3.26: Matrici di posa del robot FR3 (parte 2)

- La posa A_{start} è la posa che assume il robot FR3 nell'istante in cui il TiaGo cessa il suo movimento
- La posa A_{finish} è la posa che il robot FR3 deve assumere nella configurazione finale

Sebbene un singolo task reale richieda più traiettorie rettilinee, nella trattazione che segue si andranno a descrivere i calcoli per eseguire un solo spostamento rettilineo. La generalizzazione a più spostamenti, infatti, non aggiunge valore ai risultati, ma richiederebbe solo un algoritmo che riesca a gestire in sequenza i vari set di pose che l'end effector deve raggiungere.

Essendo richiesta una traiettoria rettilinea, si rientra nel caso di “Calcolo traiettoria in Spazio Cartesiano”. Serve quindi definire i seguenti parametri:

- Il tempo T in cui il robot deve passare dalla posa iniziale a quella finale
- Il tipo di traiettoria, che lega la variabile temporale indipendente con le tre variabili spaziali x , y e z .

Le traiettorie più comuni usate sono:

- Sinusoidale
- Polinomiale
- Trapezio di velocità

Scelta la traiettoria si ricava il valore di s in funzione del tempo ed è possibile calcolare le posizioni e le pose intermedie dell'end effector. Per quanto riguarda la posizione, il legame con la variabile adimensionale s e la variabile temporale t sono:

$$p(s) = p_{\text{start}} + s(p_{\text{finish}} - p_{\text{start}}) \quad (3.60)$$

$$p(t) = p_{\text{start}} + s(t)(p_{\text{finish}} - p_{\text{start}}) \quad (3.61)$$

In cui s definisce la traiettoria che si vuole seguire. Per definizione, s varia tra 0 e 1 quindi:

$$p(s = 0) = p_{\text{start}}, \quad p(s = 1) = p_{\text{finish}} \quad (3.62)$$

Per quanto riguarda la parte di orientazione, note la posa iniziale e finale che assumerà l'end effector, si ricava la matrice di rotazione, come segue:

$$A(s) = \text{Rot}(h, s\theta) \cdot A_{\text{start}} \quad (3.63)$$

$$A(t) = \text{Rot}(h, s(t)\theta) \cdot A_{\text{start}} \quad (3.64)$$

$$A_{\text{finish}} = \text{Rot}(h, \theta) \cdot A_{\text{start}} \quad (3.65)$$

$$\text{Rot}(h, \theta) = A_{\text{finish}} \cdot A_{\text{start}}^{-1} \quad (3.66)$$

Dalla matrice di rotazione si ricavano l'angolo θ e il versore h intorno al quale avviene la rotazione istantanea. È quindi possibile definire anche le pose intermedie:

$$A = A(s) = \text{Rot}(h, s\theta) \cdot A_{\text{start}} \quad (3.67)$$

Per ricavare gli angoli q dei giunti del robot, è sufficiente ricostruire la matrice 4×4 che rappresenta la posa generalizzata e applicare l'algoritmo di cinematica inversa di tipo geometrico []:

$$\hat{A} = \begin{bmatrix} A & p \\ 0 & 1 \end{bmatrix} \quad (3.68)$$

$$q = \text{Inverse_kinematic}(\hat{A}) \quad (3.69)$$

Nota \hat{A} in funzione di s , si può definire q in funzione di s , e quindi in funzione dt . È inoltre possibile calcolare anche le velocità angolari \dot{q} e le accelerazioni angolari \ddot{q} come:

$$\dot{q} = \frac{dq}{dt} \quad (3.70)$$

$$\ddot{q} = \frac{d\dot{q}}{dt} \quad (3.71)$$

3.7 Percorso dell'End Effector con applicazione della Collision Avoidance

Finora si è definita la traiettoria che il robot deve eseguire come successione di pose discretizzate nel tempo seguendo il "Calcolo in spazio Cartesiano" standard, mentre in questo paragrafo si è studiato un nuovo approccio, che si basa sul calcolo delle velocità $v(t)$ dell'end effector, istante per istante dove:

$$v(t) = \frac{d(p(t))}{dt} \quad (3.72)$$

Il nuovo metodo è infatti risultato indispensabile per l'applicazione della collision avoidance:

- Permette di tenere in considerazione l'effetto delle velocità repulsive sul robot FR3, indispensabile per l'applicazione della collision avoidance. Tale aspetto è approfondito nel prossimo paragrafo

- L'applicazione della collision avoidance può portare l'end effector a deviare dal percorso calcolato con il metodo classico, rendendolo di fatto inapplicabile.

Per queste ragioni, per definire il nuovo metodo di calcolo del percorso dell'end effector, si ipotizza di trovarsi in una configurazione generica, con l'end effector in una posizione che non giace necessariamente sulla retta che unisce i punti definiti dai vettori p_{start} e p_{finish} e con una posa arbitraria.

Dalle condizioni iniziali, si ricava la distanza D che separa il punto di partenza dal punto di arrivo:

$$D = \sqrt{(x_{\text{start}} - x_{\text{finish}})^2 + (y_{\text{start}} - y_{\text{End}})^2} \quad (3.73)$$

Quindi si calcola la distanza d , che separa l'end effector nella posizione nota dalla posizione che esso da raggiungere:

$$d = \sqrt{(x_{\text{EE}} - x_{\text{finish}})^2 + (y_{\text{EE}} - y_{\text{finish}})^2} \quad (3.74)$$

Avendo imposto un percorso nello spazio che muove l'end effector lungo una retta, esiste un legame tra le distanze d , D , e la variabile s . Nel dettaglio, ad ogni coppia di distanze d e D , è possibile definire un valore s_d come:

$$s_d = s(d) = \frac{D - d}{D} \quad (3.75)$$

Il valore s_d è per definizione compreso tra 0 e 1 e con esso è possibile definire in quale punto della traiettoria l'end effector si trova basandosi sulle distanze invece che sul tempo trascorso. Noto s , è quindi possibile definire anche il valore della sua derivata.

È quindi possibile calcolare v :

$$v(t) = \frac{d(p(t))}{dt} = \dot{s}(t)(p_{\text{finish}} - p_{\text{start}}) \quad (3.76)$$

Si nota, però, che questo procedimento vale solo se è noto p_{start} , cioè, se l'end effector giace sulla retta che unisce i punti definiti dai vettori p_{start} e p_{finish} definiti all'inizio della trattazione. Come viene approfondito in seguito, però, si deve presupporre che l'end effector possa occupare una posizione qualsiasi nello spazio. In base a questo, la retta su cui avviene il moto è definita dai vettori posizione \mathbf{p}^* , ovvero la posizione nota dell'end effector, e la posizione p_{finish} nota.

Per risolvere il problema, occorre definire il vettore $\mathbf{p}_{\text{start}}^*$, diverso da $\mathbf{p}_{\text{start}}$, che è calcolabile come segue.

$$\mathbf{p}_{\text{start}}^* = \frac{\mathbf{p}^* - \mathbf{p}_{\text{finish}} \cdot s}{1 - s} \quad (3.77)$$

Si può quindi calcolare la velocità $v(t)$ come:

$$v(t) = \dot{s}(t) \cdot (\mathbf{p}_{\text{finish}} - \mathbf{p}_{\text{start}}^*) \quad (3.78)$$

Nota la velocità è possibile ricavare la nuova posizione e la nuova posa dell'end effector servendosi di una discretizzazione numerica del tipo:

$$p = p(t + \Delta t) = p(t) + v(t) \cdot \Delta t \quad (3.79)$$

Un ragionamento simile va applicato per l'angolo θ . In questo caso è nota la matrice \hat{A}^* e la matrice A_{finish} che indica la posa finale che dovrà raggiungere l'end effector.

$$A_{\text{finish}} = \text{Rot}(h, \theta_{\text{interm}}) \cdot \hat{A}^* \quad (3.80)$$

$$\text{Rot}(h, \theta_{\text{interm}}) = A_{\text{finish}} \cdot \hat{A}^* - 1 \quad (3.81)$$

Si ricavano quindi h e θ_{interm} , da cui:

$$\theta = \frac{\theta_{\text{interm}}}{1 - s} \quad (3.82)$$

È quindi possibile definire la velocità angolare ω dell'end effector:

$$\omega(t) = \dot{s} \cdot \theta \cdot \hat{A}^* \cdot h \quad (3.83)$$

Nota A^* (da qui in poi chiamata semplicemente A), si ricavano gli angoli di Eulero; quindi, si calcolano i nuovi angoli di Eulero con la stessa discretizzazione temporale usata per il calcolo della posizione $p(t)$, infine si riconvertono gli angoli di Eulero nella rispettiva posa.

$$A = A(t) \implies \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} \quad (3.84)$$

$$\begin{bmatrix} \alpha(t + \Delta t) \\ \beta(t + \Delta t) \\ \gamma(t + \Delta t) \end{bmatrix} = \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \Delta t \quad (3.85)$$

$$\begin{bmatrix} \alpha(t + \Delta t) \\ \beta(t + \Delta t) \\ \gamma(t + \Delta t) \end{bmatrix} \implies \text{Calcolo Posa } A(t + \Delta t) \quad (3.86)$$

Si può quindi ricostruire la matrice di trasformazione 4×4 e calcolare gli angoli q nei giunti con l'algoritmo di cinematica inversa.

$$\hat{A} = \begin{bmatrix} A & p \\ 0 & 1 \end{bmatrix} \quad (3.87)$$

$$q = \text{Inverse_kinematic}(\hat{A}) \quad (3.88)$$

Notare inoltre che si è ricaduti nella stessa situazione di partenza e che per calcolare la nuova configurazione bisognerà ripetere gli stessi passaggi fino al raggiungimento da parte dell'end effector della posa e della posizione desiderata.

3.8 Applicazione della normativa al robot FR3

Illustrato i passaggi per il calcolo della traiettoria in spazio cartesiano secondo il nuovo metodo, si deve implementare la capacità da parte del robot di evitare ostacoli, e soprattutto di rispettare la normativa ISO/TS 15066 che indica i parametri da rispettare per

evitare l'urto tra uomo e robot. Come per il robot mobile, anche in questo caso l'obiettivo della collision avoidance è permettere al robot di lavorare senza causare problemi all'operatore umano con cui condivide la zona di lavoro, quindi bisogna supporre che la persona, come nel caso con il robot mobile, deve essere libera di muoversi liberamente nello spazio, senza vincoli di alcuna natura. La formula dettata dalla normativa è:

$$d_{\min} = (v_{\text{human}} + v_{\text{robot}}) \cdot T_r + \frac{v_{\text{robot}}^2}{2a_s} \quad (3.89)$$

L'applicazione della normativa per l'ottenimento di un algoritmo di collision avoidance presenta però notevoli differenze dal caso con il robot mobile. Questo, infatti, è vincolato a muoversi su un piano x-y e ad una quota z costante nel tempo tale da non poter interferire con il busto di una persona. Essendo inoltre le velocità relativamente alte, la distanza minima di sicurezza cresce di conseguenza, rendendo l'uso di geometrie più semplici legittimo. Nel caso di un robot antropomorfo, però, si sa che:

- la catena di link del robot che si muove in uno spazio tridimensionale
- l'end effector può operare alla stessa quota del busto della persona
- l'end effector può lavorare in prossimità degli arti superiori della persona

Non è quindi possibile assimilare il manichino ad un cilindro. Lo stesso discorso va applicato al robot FR3, che non può essere assimilato ad una forma geometrica elementare. Diventa quindi necessario trovare un nuovo metodo per identificare gli operatori e creare un'interazione efficace.

Si è quindi optato per una soluzione intermedia: nell'ambiente di simulazione sia il robot che il manichino sono stati assimilati con un serie di forme geometriche elementari tali da ricoprirli per intero [13]. Nello specifico:

- il robot è stato approssimato con 15 sfere solidali ai link del robot FR3
- il manichino è stato approssimato con una serie di sfere e cilindri solidali alle varie parti del corpo dello stesso:
 - testa, spalle, gomiti e mani sono stati approssimati con delle sfere
 - busto, avambracci e braccia sono stati approssimati con dei cilindri

Si riportano delle immagini esplicative in 3.27 e 3.28.

Avendo a che fare con 15 elementi da parte del robot e 12 elementi da parte del manichino, ci sono 180 possibili coppie di elementi di cui poter valutare la distanza. Un possibile approccio vedrebbe il considerare la coppia di elementi che sono posti alla distanza minima e applicare verificare che la normativa sia rispettata. Nel caso studiato, però, si è seguito un metodo alternativo: si osserva che ogni sfera del robot è solidale a uno specifico link, come riportato in tabella 3.1

Si considerano quindi 6 permutazioni, ciascuna da effettuare tra le sfere relative al link i -esimo e gli elementi che approssimano il manichino. Di ciascuna permutazione viene quindi considerata la coppia di elementi situati alla distanza minima. Si hanno quindi

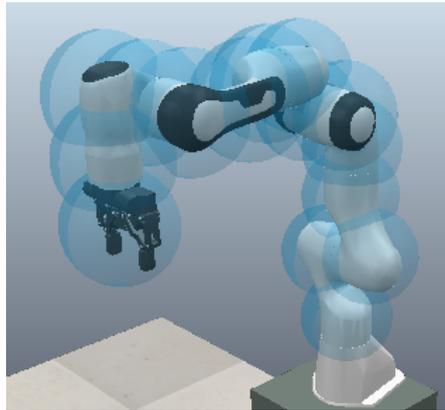


Figura 3.27: Robot FR3 approssimato da sfere



Figura 3.28: Manichino approssimato da sfere e cilindri nella parte superiore del corpo

sei coppie di elementi, ciascuna caratterizzata da due elementi posti a una distanza d l'uno dall'altro, e in ciascuna è presente un vettore λ_{coll} che giace sulla retta che unisce i baricentri dei due elementi e punta verso l'elemento del manichino.

Si deve quindi calcolare la velocità della sfera, denominata v_{sfera} , che sarà funzione di:

- q
- \dot{q}
- A_{TiaGo}
- v_{TiaGo}
- ω_{TiaGo}

Tabella 3.1: Relazione tra ciascuna sfera e il link del robot FR3 al quale sono solidali

SFERA	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LINK DI RIFERIMENTO	1	1	2	2	3	3	4	4	4	5	5	6	6	7	7

Per calcolare v_{sfera} , invece:

$$({}^{\text{TiaGo}}\hat{A})_{\text{sfera}} = \text{Forward kinematic}(q) \quad (3.90)$$

$$({}^0\hat{A})_{\text{sfera}} = ({}^0\hat{A})_{\text{TiaGo}} \cdot ({}^{\text{TiaGo}}\hat{A})_{\text{sfera}} \quad (3.91)$$

$$({}^{\text{TiaGo}}\hat{A})_{\text{sfera}} = \begin{bmatrix} ({}^0A)_{\text{sfera}} & p_{\text{sfera}} \\ 0 & 1 \end{bmatrix} \quad (3.92)$$

Si noti che il vettore p_{sfera} , ovvero il vettore posizione della sfera generica nel sistema di riferimento 0, si può calcolare la Jacobiana rispetto al TiaGo e, quindi, la velocità v_{sfera} :

$$({}^{\text{TiaGo}}\mathbf{J})_{\text{sfera}} = \text{Jacobiana}(q, \mathbf{p}_{\text{sfera}}) \quad (3.93)$$

$$v_{\text{sfera}} = v_{\text{TiaGo}} + \omega_{\text{TiaGo}} \times p_{\text{sfera}} + ({}^0A)_{\text{sfera}} \cdot ({}^{\text{TiaGo}}\mathbf{J})_{\text{sfera}} \cdot \dot{q} \quad (3.94)$$

Si calcola quindi la componente di v_{sfera} parallela a λ_{coll} :

$$v_{\text{coll}} = |v_{\text{sfera}}| (\lambda_{\text{sfera}} \cdot \lambda_{\text{coll}}) \lambda_{\text{coll}} \quad (3.95)$$

In cui:

$$\lambda_{\text{sfera}} = \frac{v_{\text{sfera}}}{|v_{\text{sfera}}|} \quad (3.96)$$

Si ricava v_{norm} , ovvero la velocità massima ammissibile parallela a λ_{coll} , applicando la formula fornita dalla normativa e utilizzando, per gli altri parametri, gli stessi valori usati per l'applicazione della normativa al TiaGo. La formula è:

$$v_{\text{norm}} = -a_s T_r - v_{\text{human}} + \sqrt{(a_s T_r)^2 + v_{\text{human}}^2 + 2a_s d} \quad (3.97)$$

v_{norm} andrà quindi confrontato con v_{coll} . Serve quindi calcolare la velocità lineare della sfera nello spazio, ovvero del punto al quale essa è solidale. Di sotto sono riportati i vari passaggi:

Sarà quindi possibile calcolare la velocità repulsiva v_{rep} che la persona esercita sul robot FR3. I risultati sono:

$$v_{\text{rep}} = \begin{cases} v_{\text{coll}} - v_{\text{norm}}, & \text{if } v_{\text{norm}} \leq |v_{\text{coll}}| \\ 0, & \text{if } v_{\text{norm}} > |v_{\text{coll}}| \end{cases} \quad (3.98)$$

Ripetendo il procedimento per tutti e 6 i link del FR3, bisogna calcolare come la velocità v dell'end effector risente dell'azione delle velocità repulsive, che agiscono su punti del robot diversi dall'end effector.

Per ognuna delle sfere su cui agisce una velocità repulsiva si devono considerare la sotto-matrice di ${}^{\text{TiaGo}}\mathbf{J}_{\text{sfera}}$, costituita dalle sue prime tre righe, denominata $\mathbf{J}_{\text{sfera}_v}$:

$${}^{\text{TiaGo}}\mathbf{J}_{\text{sfera}} = \begin{bmatrix} \mathbf{J}_{\text{sfera}_v} \\ \mathbf{J}_{\text{sfera}_\omega} \end{bmatrix} \quad (3.99)$$

Noti $\mathbf{J}_{\text{sfera}_v}$ e v_{rep} , si ricava $\dot{q}_{\text{rep sfera}}$, ovvero le velocità angolari dei giunti a seguito dell'azione repulsiva:

$$\dot{q}_{\text{rep sfera}} = \begin{bmatrix} \mathbf{J}_{\text{sfera}_{11}} & \dots & \mathbf{J}_{\text{sfera}_{16}} \\ \mathbf{J}_{\text{sfera}_{21}} & \dots & \mathbf{J}_{\text{sfera}_{26}} \\ \mathbf{J}_{\text{sfera}_{31}} & \dots & \mathbf{J}_{\text{sfera}_{36}} \end{bmatrix}^{-1} \cdot \mathbf{v}_{\text{rep}} \quad (3.100)$$

Si noti che $\mathbf{J}_{\text{sfera}_v}$ è una matrice di dimensioni 3×6 , quindi non invertibile. Si adotta quindi il metodo della matrice pseudo-inversa, secondo cui:

$$(\mathbf{J}_{\text{sfera}_v})^{-1} = \left(\left((\mathbf{J}_{\text{sfera}_v})^T \cdot \mathbf{J}_{\text{sfera}_v} \right) \right)^{-1} \cdot (\mathbf{J}_{\text{sfera}_v})^T \quad (3.101)$$

Tuttavia, $\mathbf{J}_{\text{sfera}_v}$, per definizione stessa di Jacobiana, può avere un certo numero di colonne nulle, che dipende dalla sfera che si sta considerando. Più precisamente dipende dal numero di giunti che contribuiscono al suo movimento. Ne consegue che, meno giunti ci sono a movimentare la sfera analizzata, maggiori sono le probabilità che una singolarità renda la matrice $\mathbf{J}_{\text{sfera}_v}$ non invertibile anche con il metodo della pseudo-inversa. In questo caso si è adottata la formula sperimentale del tipo:

$$(\mathbf{J}_{\text{sfera}_v})^{-1} = C (\mathbf{J}_{\text{sfera}_v})^T \quad (3.102)$$

in cui C è un coefficiente ricavato sperimentalmente. In seguito a ulteriori test, per il calcolo della matrice $\mathbf{J}_{\text{sfera}_v}^{-1}$, si è scelto di applicare la seguente regola:

- Per sfere solidali ai link che variano dal link 5 al link 7 si calcola la pseudo-inversa della sottomatrice $\mathbf{J}_{\text{sfera}_v}$
- Per sfere solidali ai link che variano dal link 1 al link 4 si calcola la matrice inversa moltiplicando la trasposta di $\mathbf{J}_{\text{sfera}_v}$ per un coefficiente C ricavato in un esperimento simile.

Applicando il metodo a tutte le sfere solidali al robot, si ottiene l'effetto complessivo che l'azione repulsiva esercita sui giunti in termine di velocità angolari $\dot{q}_{\text{rep sfera } j}$. Si può quindi ricavare quanto vale la velocità dell'end effector comprensivo dell'azione repulsiva [22].

$$v_{\text{EE TOT}} = v_{\text{EE}} + \mathbf{J}_{\text{EE}} \sum_{j=1}^6 \dot{q}_{\text{rep sfera } j} \quad (3.103)$$

in cui:

- v_{EE} è la velocità dell'end effector senza l'effetto repulsivo

- J_{EE} è la Jacobiana che fa riferimento alla posizione dell'end effector.

La velocità $v_{EE\ TOT}$ sarà la velocità dell'end effector effettiva, con cui calcolare la nuova posizione dell'end effector con la discretizzazione numerica. Notare che, in un caso di distanza uomo-robot, la velocità repulsiva “sentita” dall'end-effector può risultare molto alta e quindi non raggiungibile con celerità dal robot reale o addirittura richiedere ai giunti velocità angolari che superano quelle ammissibili da catalogo. Visto che un robot collaborativo deve essere in grado di arrestarsi in condizioni di emergenza, si impone che il robot arresti il suo movimento quando $v_{EE\ TOT}$ supera un valore limite.

Capitolo 4

Risultati

Nel Capitolo 2 si è sviluppato un metodo per identificare la sequenza di smontaggio ottimale della batteria, mentre nel Capitolo 3 si è illustrato l'algoritmo che permette ai due operatori di interagire tra loro. In questo capitolo vengono applicati i due algoritmi per osservare l'interazione tra uomo e robot e l'impatto che si ha sullo smontaggio della batteria.

Come osservato in 2.19, per i dati scelti, i tempi ciclo ricavati dall'algoritmo di ottimizzazione della sequenza di smontaggio della batteria differiscono di poco tra loro. Si è quindi deciso di scegliere pochi casi specifici e studiare in quanti casi vi è interazione tra uomo e robot e quali sono le loro proprietà principali. Sono state scelte due sequenze di smontaggio:

- Sequenza di smontaggio 1. Tempo ciclo: 3625 secondi
- Sequenza di smontaggio 2. Tempo ciclo: 3627 secondi

Per entrambe le sequenze si vuole studiare:

1. quando i due operatori lavorano in zone vicine
2. quanto volte queste situazioni si ripetono nel corso dell'intera sequenza di smontaggio
3. come si differenziano tra di loro
4. l'impatto che ha la distanza sulle operazioni
5. il rispetto delle distanze di sicurezza

In 4.1 e 4.2 sono riportate le sequenze di smontaggio espresse con i Diagrammi di Gantt ed è stato associato un colore ad ogni intervallo di tempo, in base alle sue proprietà:

- tempo morto: intervallo di tempo in cui l'operatore non effettua alcuna operazione. Bianco
- tempo di cambio utensile: intervallo di tempo in cui l'operatore sostituisce l'utensile per adattarsi al prossimo task. Blu

- tempo di spostamento: intervallo di tempo necessario per spostarsi nella zona su cui lavorare per effettuare il task successivo. Giallo
- tempo utile: intervallo di tempo ciclo di svolgimento del task, come riportato dalla tabella Excel (come spiegato nel Capitolo 2). Verde
- tempo di interferenza: intervallo di tempo aggiuntivo necessario all'operatore per svolgere il task assegnato. Rosso

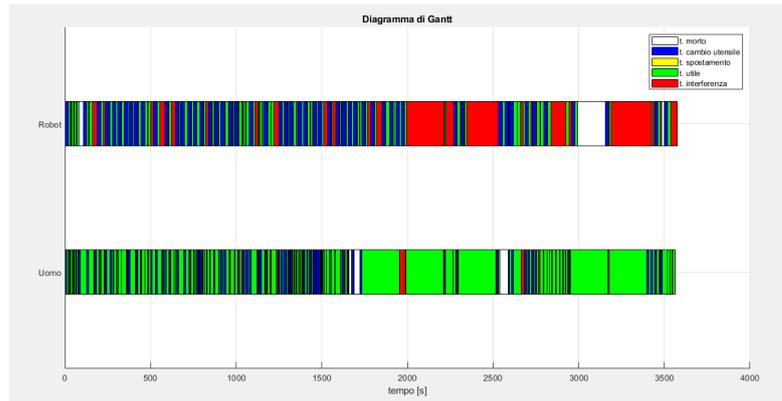


Figura 4.1: Diagramma di Gantt della sequenza di smontaggio 1, tempo ciclo di 3625 s

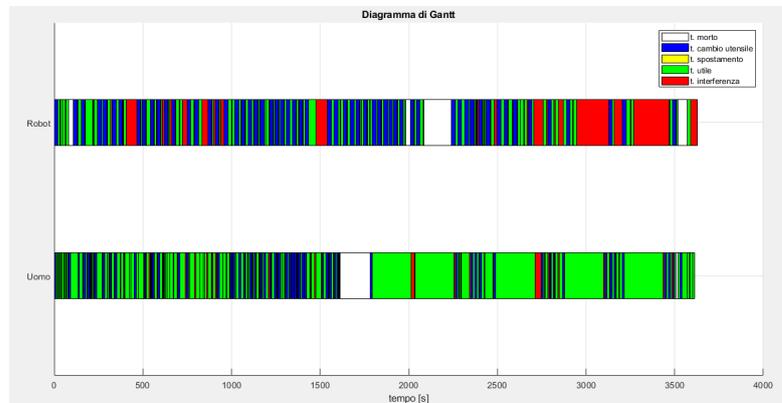


Figura 4.2: Diagramma di Gantt della sequenza di smontaggio 2, tempo ciclo di 3627 s

Nota la simulazione si possono ricavare diversi casi in cui uomo e robot operano a distanza ravvicinata. In questi intervalli di tempo emergono le interazioni tra uomo e robot e sono caratterizzati da:

- una coppia di task, uno svolto dall'uomo e uno svolto dal robot
- una coppia di utensili, uno per ciascun operatore

- l'intervallo di tempo in cui i due task avvengono contemporaneamente
- l'indice f_v che determina quanto deve rallentare il robot

Non sono stati ritenuti rilevanti invece:

- il tipo di task eseguito
- l'utensile utilizzato
- l'intervallo di tempo in cui i due operatori lavorano vicini come ricavato dall'algoritmo

Se per i primi due punti è evidente la scarsa rilevanza nella collision avoidance, per il terzo può sembrare più difficoltoso da trascurare. La scelta deriva dall'incertezza del rispetto dei tempi dettati dalla simulazione durante l'esecuzione dei vari task. Sommando a ciò la presenza di variabili imprevedibili in fase di progetto, si può dire che non si potrà mai avere la certezza che la sequenza di smontaggio trovata verrà rispettata secondo i tempi prestabiliti. È quindi lecito supporre che due task in teoria eseguiti in brevissimo tempo possano richiedere più tempo del dovuto, permettendo di mettere sullo stesso livello casi in cui l'interazione uomo-robot dura pochi secondi con altri casi in cui l'interazione può anche superare il minuto.

Nello studio delle casistiche, per concentrarsi soprattutto sul comportamento del robot, ci si è soffermati su casi in cui il robot si muove nello spazio mentre la persona rimane ferma. Come ulteriore semplificazione al problema, l'intera esecuzione del task relativo alla persona è stata assimilata al movimento del braccio destro che, a intervalli di tempo regolari, si allontana dal fianco e si distende orizzontalmente, raggiungendo un punto nello spazio posto al di sopra della batteria.

4.1 Casistica 1

La prima casistica è prelevata dalla sequenza di smontaggio n°1 ed ha le seguenti proprietà:

- Il TiaGo si muove mentre la persona rimane ferma.
- Il TiaGo rischia di entrare nello stesso spazio di lavoro della persona. Questa informazione indica l'impatto che ha l'algoritmo di collision avoidance del TiaGo nel percorso che quest'ultimo dovrà compiere.
- Il robot FR3 non lavora nelle vicinanze della persona.

In 4.3 e 4.4 è riportata una rappresentazione schematica della zona di lavoro nell'istante iniziale e nell'istante finale. Per il caso rappresentativo della casistica 1, l'operatore composto dai due robot dovrà svolgere le seguenti operazioni:

- Scegliere quale direzione lo porterà a destinazione nel minor tempo possibile (al netto dell'eventualità di dover allungare il percorso per via del manichino).
- In base a tale scelta, ruotare di $\pm 90^\circ$ il TiaGo.

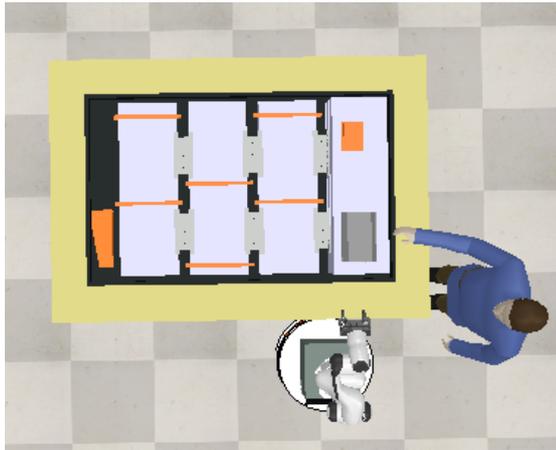


Figura 4.3: Posizione dei due operatori ad inizio simulazione

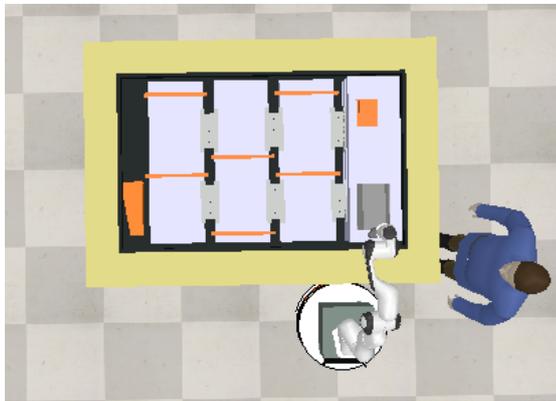


Figura 4.4: Posizione dei due operatori a simulazione conclusa

- Costeggiare il tavolo, mantenendo una distanza di sicurezza dal manichino, fino a quando non raggiunge il punto prefissato.
- Orientare il TiaGo in modo adeguato.
- Orientare il robot FR3 secondo la posizione e la posa richiesta.

Inizialmente il TiaGo rileva la distanza che lo separa dal Tavolo; quindi, si orienta in modo tale da non poterlo urtare. Il robot mobile, quindi, avanza fino a che non si avvicina al manichino, momento in cui interviene l'algoritmo di collision avoidance del robot mobile. Dal momento che il manichino rimane confinato nel proprio spazio di lavoro, il TiaGo gli gira intorno. Il percorso di aggiramento dell'ostacolo causato dal manichino è eseguito mantenendo costante la distanza di sicurezza, così da rispettare la normativa ISO/TS 15066 e allo stesso tempo ridurre i tempi di spostamento al minimo. Il risultato è che il percorso assumerà in parte l'aspetto di un arco di circonferenza. Nel momento in cui il TiaGo non è più obbligato ad aggirare il manichino, il percorso cessa di essere di

tipo circolare. Ciò avviene quando il solo intervento della velocità v_{base} è sufficiente ad allontanare il TiaGo dal manichino. Da qui il robot mobile, evitando di urtare il tavolo, raggiunge la posizione e l'orientamento richiesto.

Per quanto riguarda l'algoritmo di collision avoidance legato al robot FR3, non è possibile trarre alcun dato rilevante: gli spazi di lavoro dei due operatori sono sufficientemente lontani, tanto da permettere all'end effector del robot FR3 di raggiungere la posa desiderata rispettando il tempo T scelto a monte, seguendo il percorso rettilineo preimpostato. Per

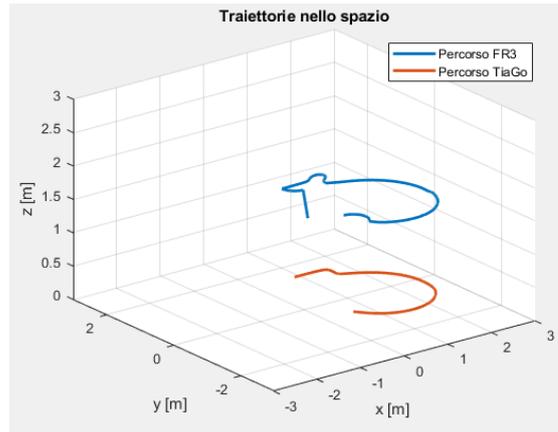


Figura 4.5: Percorsi del TiaGo e dell'end effector del FR3 nello spazio cartesiano

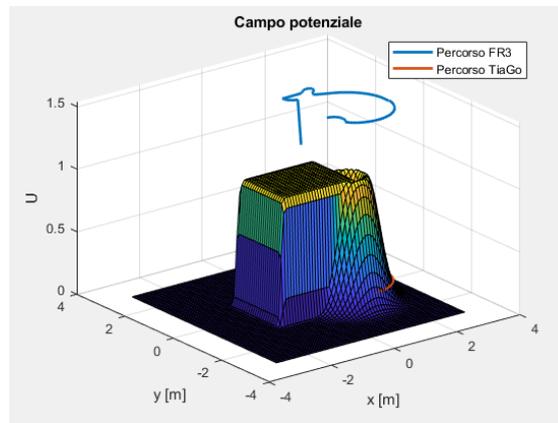


Figura 4.6: Campi Potenziali repulsivi

verificare il rispetto della normativa si considera, per ogni istante di tempo, l'elemento sfera, solidale al robot, che si trova alla minima distanza d dal robot, con d pari a:

$$d = \sqrt{(x_{\text{sfera}} - x_{\text{Human element}})^2 + (y_{\text{sfera}} - y_{\text{Human element}})^2} - r_{\text{sfera}} - r_{\text{Human element}} \quad (4.1)$$

in cui:

- $x_{\text{Human element}}$ è la coordinata lungo l'ascissa del vettore posizione della sfera o del cilindro del manichino che è più vicino al robot FR3
- $y_{\text{Human element}}$ è la coordinata lungo l'ordinata del vettore posizione della sfera o del cilindro del manichino che è più vicino al robot FR3
- r_{sfera} è il raggio della sfera che è più vicino al manichino
- $r_{\text{Human element}}$ è il raggio della sfera o del cilindro che è più vicino al robot FR3

Per ciascuna sfera si considera la distanza d e la velocità $|\mathbf{v}_{\text{avanz}}|$, che è pari a:

$$|v_{\text{avanz}}| = |v_{\text{coll}}| - |v_{\text{rep}}| \quad (4.2)$$

dove $|\mathbf{v}_{\text{coll}}|$ e $|\mathbf{v}_{\text{rep}}|$ sono i moduli delle velocità calcolate nel Capitolo 3, penultimo paragrafo. $|\mathbf{v}_{\text{avanz}}|$ viene confrontata graficamente con $|\mathbf{v}_{\text{norm}}|$, calcolata dalla normativa usando la distanza d . Essendo $|\mathbf{v}_{\text{rep}}|$ sempre nulla in questa casistica, tutti i valori $|\mathbf{v}_{\text{avanz}}|$ sono al

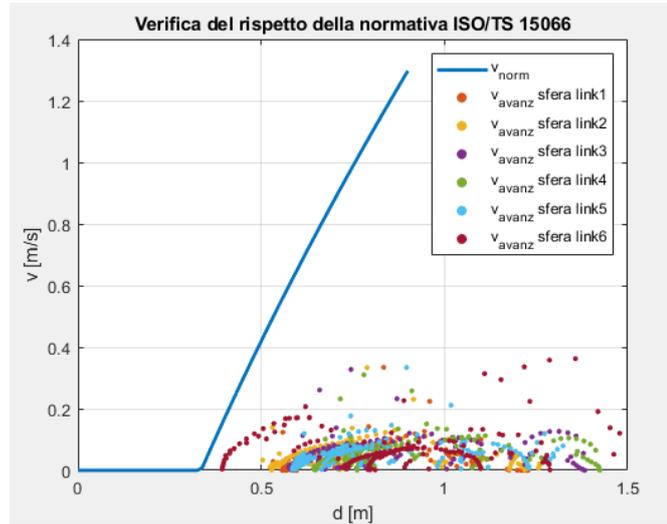


Figura 4.7: Confronto tra le velocità $|\mathbf{v}_{\text{avanz}}|$ delle sfere solidali al robot FR3 e la curva \mathbf{V}_{norm}

di sotto della curva blu, che rappresenta la velocità $|\mathbf{v}_{\text{norm}}|$ in funzione di d . Ciò indica che le velocità dei link del robot sono rispettate durante l'intero spostamento e anche durante il cambio di posa del robot FR3.

Si applica lo stesso ragionamento anche al TiaGo, verificando che mantenga dal manichino una distanza coerente con la normativa.

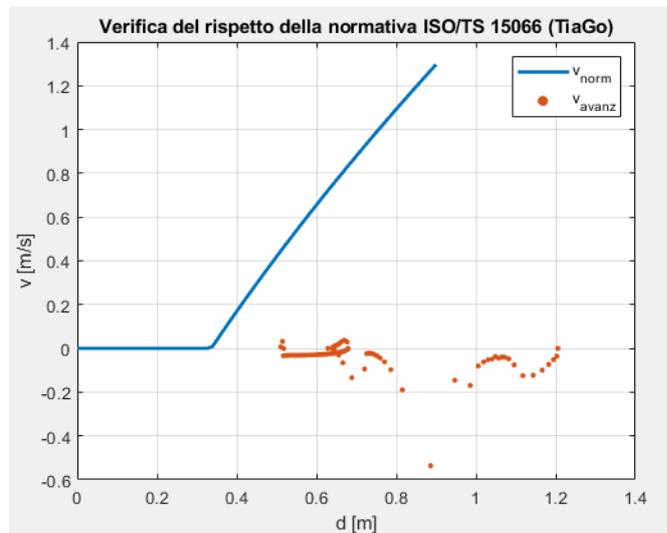


Figura 4.8: Confronto tra le velocità $|v_{avanz}|$ del TiaGo e la curva v_{norm}

4.2 Casistica 2

La seconda casistica scelta per lo studio ha le seguenti proprietà:

- Il TiaGo e la persona non si muovono dalla loro posizione
- • Il robot FR3 entra nello stesso spazio di lavoro della persona. L'algoritmo di collision avoidance del FR3 ha quindi un impatto nel percorso che quest'ultimo dovrà compiere

In figura 4.9 è riportata una rappresentazione schematica della zona di lavoro nell'istante iniziale e nell'istante finale. Per il caso scelto FR3 dovrà solamente orientare il robot

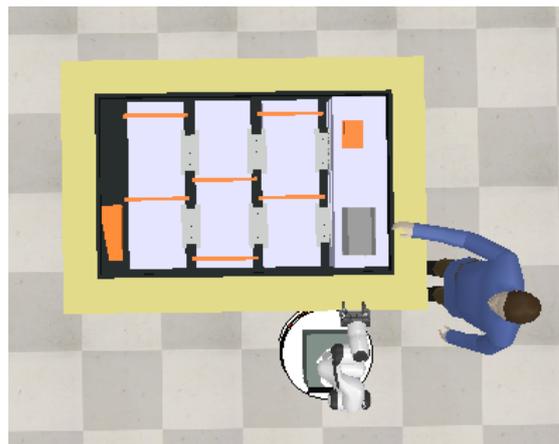


Figura 4.9: Posizione dei due operatori ad inizio simulazione

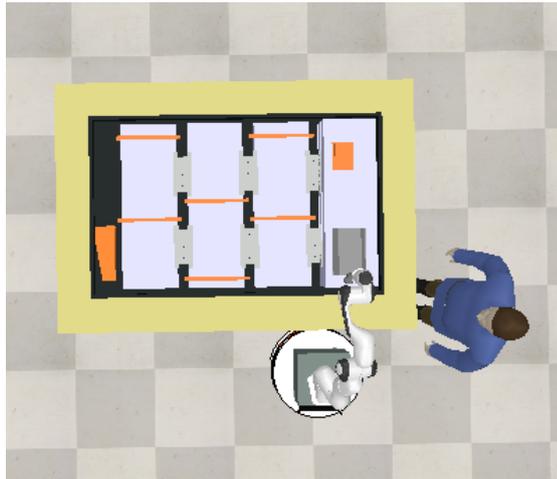


Figura 4.10: Posizione dei due operatori a simulazione conclusa

antropomorfo secondo la posa richiesta, garantendo costantemente la distanza di sicurezza dall'uomo imposta dalla normativa.

Diversamente dalla casistica precedente, qui l'algoritmo di collision avoidance ha un ruolo fondamentale nel definire le pose del robot FR3: gli spazi di lavoro dei due operatori sono molto vicini, e il rischio di urto in assenza di un algoritmo di prevenzione sarebbe molto alto. L'end effector quindi, dopo aver seguito inizialmente il percorso desiderato, subisce una deviazione dovuta alla persona, che con la mano destra occupa il suo spazio di lavoro. Quando la persona ritrae la mano l'end effector del robot FR3 è libero di raggiungere la posa desiderata con uno spostamento rettilineo, seguendo la traiettoria di riferimento.

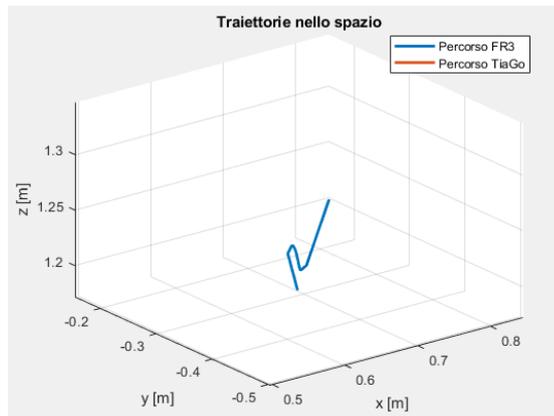


Figura 4.11: Percorsi del TiaGo e dell'end effector del FR3 nello spazio cartesiano

Anche in questo caso si verifica il rispetto della normativa confrontando, per ogni istante di tempo, l'elemento sfera, solidale al robot, che si trova alla minima distanza d dal robot e si calcolano la distanza d e la velocità $|\mathbf{v}_{avanz}|$. In questo caso, però, si riscontrano situazioni

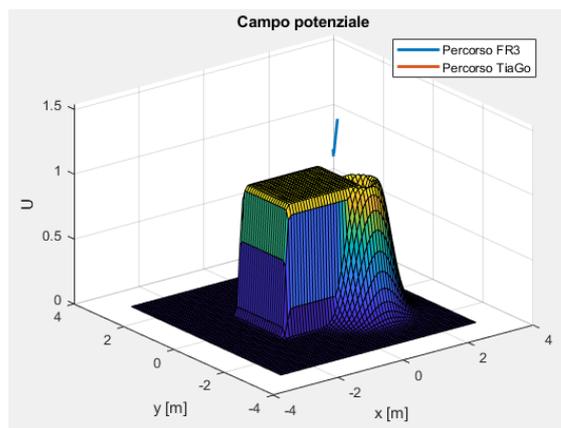
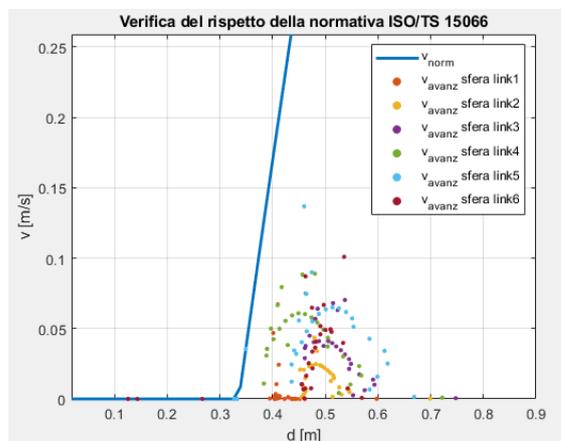


Figura 4.12: Campi Potenziali repulsivi

in cui la velocità $|\mathbf{v}_{\text{rep}}|$ non è nulla, il che indica che l'algoritmo di collision avoidance si è rivelato determinante nel garantire il rispetto della normativa nell'applicazione del robot FR3. Come risultato si avrà che:

$$\mathbf{v}_{\text{avanz}} = \mathbf{v}_{\text{norm}} \quad (4.3)$$

Graficamente, si osserva che alcuni punti, invece di trovarsi nell'area sottesa alla curva limite imposta dalla normativa, si trovano sulla curva stessa. Se, data una certa distanza d , si ha una velocità $|\mathbf{v}_{\text{avanz}}|$ che cade sulla curva limite, si deduce che la corrispettiva $|\mathbf{v}_{\text{coll}}|$ non rispetta la normativa. Notare che, in presenza di distanze ancora minori, la velocità

Figura 4.13: Confronto tra le velocità $|v_{\text{avanz}}|$ delle sfere solidali al robot FR3 e la curva \mathbf{V}_{norm} .

$|\mathbf{v}_{\text{rep}}|$ imposta a una sfera può raggiungere valori molto elevati, fino a rendere la collision avoidance impraticabile in un'applicazione reale del robot FR3. Da qui la necessità di imporre, in caso di necessità, un arresto di sicurezza nel robot. In questa situazione

il robot FR3, non compiendo più alcun movimento, diventa incapace di arrecare danni all'uomo. Tale eventualità può verificarsi in presenza di:

- Tempo di esecuzione della traiettoria basso
- Tempo di reazione del robot FR3 alto
- Movimenti bruschi da parte dell'operatore

4.3 Casistica 3

La terza casistica scelta per lo studio ha le seguenti proprietà:

- Il robot si muove mentre la persona rimane ferma.
- Il robot rischia di entrare nello stesso spazio di lavoro della persona. Questa informazione indica l'impatto che ha l'algoritmo di collision avoidance del TiaGo nel percorso che quest'ultimo dovrà compiere.
- Il robot FR3 si avvicina allo spazio di lavoro della persona. L'algoritmo di collision avoidance del FR3 ha quindi un impatto nel percorso che quest'ultimo dovrà compiere.

Si nota che la casistica 3 è una combinazione delle casistiche 1 e 2.

Di seguito è riportata una rappresentazione schematica della zona di lavoro nell'istante iniziale e nell'istante finale. Per il caso scelto l'operatore composto dai 2 robot dovrà

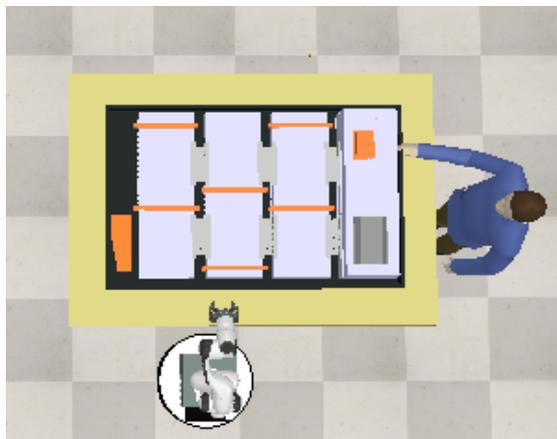


Figura 4.14: Posizione dei due operatori ad inizio simulazione

svolgere le seguenti operazioni:

- Scegliere quale direzione lo porterà a destinazione nel minor tempo possibile (al netto della presenza dell'uomo lungo il cammino)
- Ruotare di $\pm 90^\circ$ in base a tale scelta

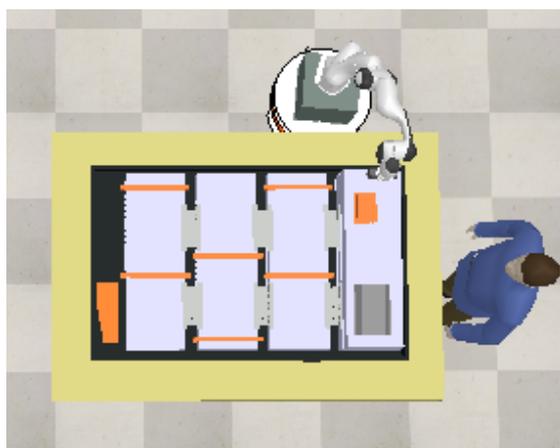


Figura 4.15: Posizione dei due operatori a simulazione conclusa

- Costeggiare il tavolo fino a quando non potrà raggiungere il punto prefissato
- Raggiunto il punto prefissato, orientare il robot antropomorfo secondo la posa richiesta

Inizialmente il TiaGo rileva la distanza che lo separa dal Tavolo; quindi, si orienta in modo tale da non poterlo urtare. Il robot mobile, quindi, avanza fino a che non si avvicina al manichino, momento in cui interviene l'algoritmo di collision avoidance del robot mobile. Dal momento che il manichino rimane confinato nel proprio spazio di lavoro, il TiaGo gli gira intorno. Il percorso di aggiramento dell'ostacolo causato dal manichino è eseguito mantenendo costante la distanza di sicurezza, così da rispettare la normativa ISO/TS 15066 e allo stesso tempo ridurre i tempi di spostamento al minimo. Il risultato è che il percorso assumerà in parte l'aspetto di un arco di circonferenza. Nel momento in cui il TiaGo non è più obbligato ad aggirare il manichino, il percorso cessa di essere di tipo circolare. Ciò avviene quando il solo intervento della velocità $|\mathbf{v}_{\text{base}}|$ è sufficiente ad allontanare il TiaGo dal manichino. Da qui il robot mobile, evitando di urtare il tavolo, raggiunge la posizione e l'orientamento richiesto.

Come nella casistica precedente, anche qui l'algoritmo di collision avoidance ha un ruolo fondamentale nel definire le pose del robot FR3: gli spazi di lavoro dei due operatori sono molto vicini, e il rischio di urto in assenza di un algoritmo di prevenzione sarebbe molto alto. L'end effector quindi, dopo aver seguito inizialmente il percorso desiderato, subisce una deviazione dovuta alla persona, che con la mano destra occupa il suo spazio di lavoro. Quando la persona ritrae la mano l'end effector del robot FR3 è libero di raggiungere la posa desiderata con uno spostamento rettilineo, seguendo la traiettoria di riferimento. Anche in questo caso si verifica il rispetto della normativa confrontando, per ogni istante di tempo, l'elemento sfera, solidale al robot, che si trova alla minima distanza d dal robot e si calcolano la distanza d e la velocità $|\mathbf{v}_{\text{avanz}}|$. Anche in questa casistica, si riscontrano situazioni in cui la velocità $|v_{\text{rep}}|$ non è nulla, il che indica che l'algoritmo di collision avoidance si è rivelato determinante nel garantire il rispetto della normativa nell'applicazione del robot FR3. I risultati sono analoghi a quelli riscontrati

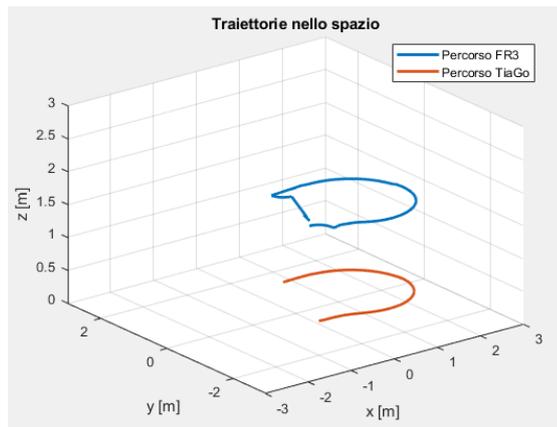


Figura 4.16: Percorsi del TiaGo e dell'end effector del FR3 nello spazio cartesiano

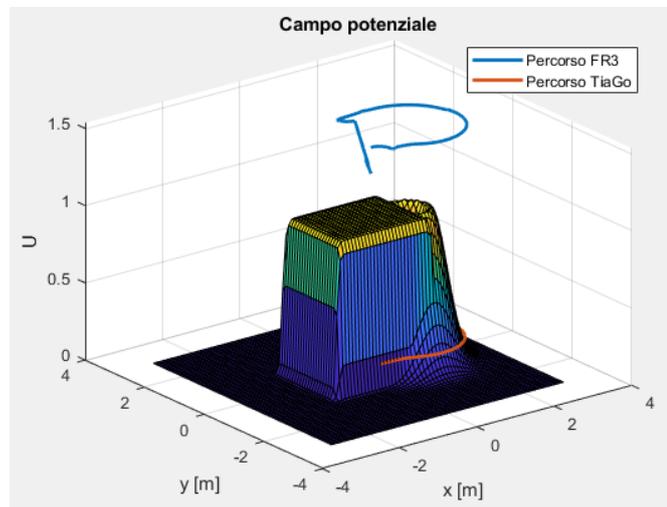


Figura 4.17: Campi Potenziali repulsivi

nella casistica 2. Anche in questa casistica, inoltre, la velocità $|\mathbf{v}_{\text{rep}}|$ imposta a una sfera può raggiungere valori molto elevati, fino a rendere la collision avoidance impraticabile in un'applicazione reale del robot FR3. Da qui la necessità di imporre, in caso di necessità, un arresto di sicurezza nel robot. In questa situazione il robot FR3, non compiendo più alcun movimento, diventa incapace di arrecare danni all'uomo.

Si applica lo stesso ragionamento anche al TiaGo, verificando che mantenga dal manichino una distanza coerente con la normativa.

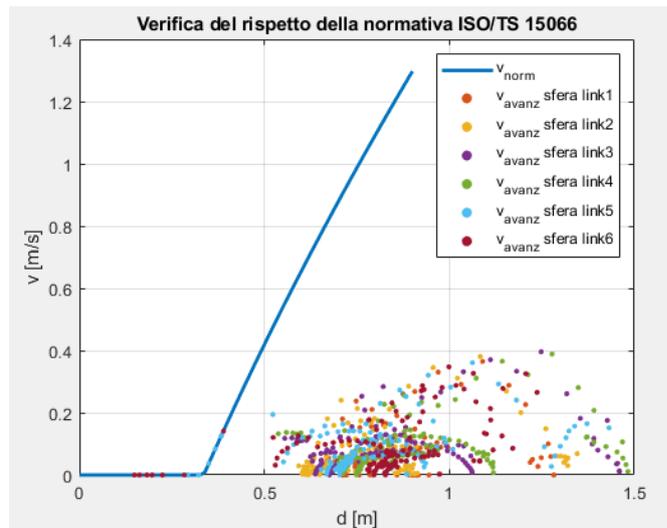


Figura 4.18: Confronto tra le velocità $|v_{avanz}|$ delle sfere solidali al robot FR3 e la curva v_{norm} .

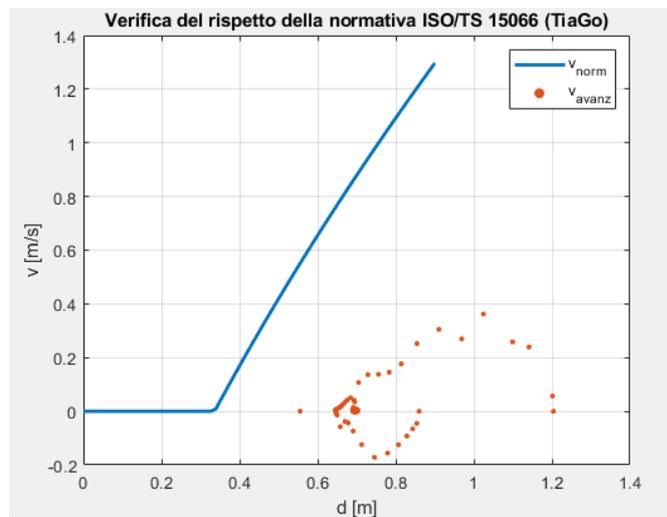


Figura 4.19: Confronto tra le velocità $|v_{avanz}|$ del TiaGo e la curva v_{norm} .

Capitolo 5

Conclusioni

In questo elaborato, si è approfondito il concetto di robot collaborativo, studiando le categorie di robot collaborativo antropomorfo e di robot collaborativo mobile, concentrandosi principalmente sugli AMV (Autonomous Mobile Robot). Allo scopo di provare che questi robot possano essere impiegati nello smontaggio di una batteria di un'auto elettrica, si è approfondito anche l'argomento delle EVBs (Electrical Vehicle Batteries). Quindi, sono stati scelti tre esempi per ciascuna categoria: come il robot antropomorfo si è scelto un Franka Emika FR3, per il robot mobile un TiaGo Base e come batteria quella impiegata nella Chevrolet Bolt 2017. Nell'applicazione il robot TiaGo si muove nel piano e sopra ad esso è montato il robot FR3 che esegue appositi task. Sono stati esaminati dei metodi per eseguire uno smontaggio della batteria con un solo operatore; dunque, è stato realizzato un algoritmo sul software Matlab per ottimizzare la sequenza di smontaggio tramite minimizzazione del tempo ciclo e ipotizzando che l'uomo e i robot lavorino in contemporanea nello stesso spazio di lavoro. Si è quindi definito un secondo algoritmo, sempre realizzato su Matlab, che impedisca ai robot di urtare la persona, per poi visualizzare i movimenti dei due operatori riproducendo i loro movimenti nell'ambiente di simulazione Coppelasim. Quindi, sono stati applicati i due algoritmi, dai quali sono stati ricavate delle casistiche in cui osservare e studiare come i due robot modificano il loro percorso in modo da non urtare l'uomo.

In una futura applicazione è possibile registrare, sempre tramite l'uso delle telecamere Kinect, movimenti più complessi, così che il manichino occupi un volume di lavoro maggiore o passi da una zona di lavoro ad un'altra. In questo modo è possibile osservare quali movimenti i due robot mobile eseguiranno per rispettare le distanze di sicurezza. L'algoritmo che implementa la collision avoidance nei due robot può inoltre essere impiegato in un'applicazione in laboratorio, dove si potrà osservare come i robot reali rispondono ai valori di set prodotti dall'algoritmo ed eventualmente migliorare le leggi di controllo sulla velocità che, in questo lavoro, sono state ricavate da test effettuati su un software che riproduce un ambiente di simulazione virtuale.

Bibliografia

- [1] Mary B Alatis e Gerhard P Hancke. “A review on challenges of autonomous mobile robot and sensor fusion methods”. In: *IEEE Access* 8 (2020), pp. 39830–39846.
- [2] Sebastian Blankemeyer et al. “Investigation of the potential for an automated disassembly process of BEV batteries”. In: *Procedia CIRP* 98 (2021), pp. 559–564.
- [3] Christoph Byner, Björn Matthias e Hao Ding. “Dynamic speed and separation monitoring for collaborative robot applications—concepts and performance”. In: *Robotics and Computer-Integrated Manufacturing* 58 (2019), pp. 239–252.
- [4] Elvira Chebotareva et al. “Basic user interaction features for human-following cargo robot TIAGo base”. In: *2020 13th International Conference on Developments in eSystems Engineering (DeSE)*. IEEE. 2020, pp. 206–211.
- [5] B Gopalakrishnan, S Tirunellayi e R Todkar. “Design and development of an autonomous mobile smart vehicle: a mechatronics application”. In: *Mechatronics* 14.5 (2004), pp. 491–514.
- [6] Yanhao He e Steven Liu. “Analytical inverse kinematics for franka emika panda—a geometrical solver for 7-dof manipulators with unconventional design”. In: *2021 9th International Conference on Control, Mechatronics and Automation (ICCMA)*. IEEE. 2021, pp. 194–199.
- [7] Jan F Hellmuth, Nicholas M DiFilippo e Musa K Jouaneh. “Assessment of the automation potential of electric vehicle battery disassembly”. In: *Journal of Manufacturing Systems* 59 (2021), pp. 398–412.
- [8] Christoph Herrmann et al. “Assessment of automation potentials for the disassembly of automotive lithium ion battery systems”. In: *Leveraging Technology for a Sustainable World: Proceedings of the 19th CIRP Conference on Life Cycle Engineering, University of California at Berkeley, Berkeley, USA, May 23-25, 2012*. Springer. 2012, pp. 149–154.
- [9] Karla L Hoffman, Manfred Padberg, Giovanni Rinaldi et al. “Traveling salesman problem”. In: *Encyclopedia of operations research and management science* 1 (2013), pp. 1573–1578.
- [10] Elif Kongar e Surendra M Gupta. “Disassembly sequencing using genetic algorithm”. In: *The International Journal of Advanced Manufacturing Technology* 30 (2006), pp. 497–506.
- [11] Laura Lander et al. “Breaking it down: A techno-economic assessment of the impact of battery pack design on disassembly costs”. In: *Applied Energy* 331 (2023), p. 120437.
- [12] Ming Li, Andrija Milojević e Heikki Handroos. “Robotics in manufacturing—The past and the present”. In: *Technical, Economic and Societal Effects of Manufacturing 4.0: Automation, Adaption and Manufacturing in Finland and Beyond* (2020), pp. 85–95.

- [13] Matteo Melchiorre et al. “A Novel Constrained Trajectory Planner for Safe Human-robot Collaboration”. In: *Proceedings of the ICINCO*. 2022.
- [14] Matteo Melchiorre et al. “Experiments on the Artificial Potential Field with Local Attractors for Mobile Robot Navigation”. In: *Robotics* 12.3 (2023), p. 81.
- [15] Matteo Melchiorre et al. “Robot collision avoidance based on artificial potential field with local attractors”. In: *Proceedings of the ICINCO*. 2022.
- [16] Matteo Melchiorre et al. “Vision-based control architecture for human–robot hand-over applications”. In: *Asian Journal of Control* 23.1 (2021), pp. 105–117.
- [17] Jordi Pages, Luca Marchionni e Francesco Ferro. “Tiago: the modular robot that adapts to different research needs”. In: *International workshop on robot modularity, IROS*. Vol. 290. 2016.
- [18] Pierpaolo Palmieri et al. “Human arm motion tracking by Kinect sensor using Kalman filter for collaborative robotics”. In: *Advances in Italian Mechanism Science: Proceedings of the 3rd International Conference of IFToMM Italy 3*. Springer. 2021, pp. 326–334.
- [19] Jing Ren et al. “A potential field model using generalized sigmoid functions”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37.2 (2007), pp. 477–484.
- [20] Francisco Rubio, Francisco Valero e Carlos Llopis-Albert. “A review of mobile robots: Concepts, methods, theoretical framework, and applications”. In: *International Journal of Advanced Robotic Systems* 16.2 (2019), p. 1729881419839596.
- [21] Roy E Sandlin Jr et al. *An Analysis of AN/FPS-16 Radar and AN/GMD-1B Rawinsonde Data Differences*. 1964.
- [22] Leonardo Sabatino Scimmi et al. “A practical and effective layout for a safe human-robot collaborative assembly task”. In: *Applied Sciences* 11.4 (2021), p. 1763.
- [23] Kathrin Wegener et al. “Disassembly of electric vehicle batteries using the example of the Audi Q5 hybrid system”. In: *Procedia Cirp* 23 (2014), pp. 155–160.
- [24] Wei-Chang Yeh. “Optimization of the disassembly sequencing problem on the basis of self-adaptive simplified swarm optimization”. In: *IEEE transactions on systems, man, and cybernetics-part A: systems and humans* 42.1 (2011), pp. 250–261.
- [25] Hai Jun Yu et al. “Trial study on EV battery recycling standardization development”. In: *Advanced Materials Research* 610 (2013), pp. 2170–2173.

Sitografia

- [26] *Agenda Digitale*. URL: <https://www.agendadigitale.eu/sanita/i-robot-entrano-in-ospedale-e-ci-aiutano-contro-il-codiv/>.
- [27] *Franka-Emika-Panda*. URL: https://www.mybotshop.de/Franka-Emika-Panda-FCI-Licence_1.

- [28] *Insideevs*. URL: <https://insideevs.it/features/580047/batterie-auto-elettriche-i-segreti/>.
- [29] *Jacobian*. URL: <https://www.rosroboticslearning.com/jacobian>.
- [30] *Universal Robots*. URL: <https://www.universal-robots.com/it/robot-collaborativi-o-cobot-cosa-sono-la-guida-definitiva/>.
- [31] *Video*. URL: https://www.youtube.com/watch?v=ssU2mjiNi_Q.

Dati di Input sequenza di smontaggio ottimale

Figura 1: Dati di input nel file Excel (parte 1)

Codice	x0	y0	Length_x	Length_y	Operatore	Tool human	Tool robot	t ciclo human	t ciclo robot	REQUIREMENTS												
B01	90	750	180	500	robot&human	Screwdriver B0	Screwdriver B0	8	12													
B02	355	955	350	90	robot&human	Screwdriver B0	Screwdriver B0	8	12													
B03	705	955	350	90	robot&human	Screwdriver B0	Screwdriver B0	8	12													
B04	1055	955	350	90	robot&human	Screwdriver B0	Screwdriver B0	8	12													
B05	1415	750	370	500	robot&human	Screwdriver B0	Screwdriver B0	8	12													
B06	90	250	180	500	robot&human	Screwdriver B0	Screwdriver B0	8	12													
B07	355	45	350	90	robot&human	Screwdriver B0	Screwdriver B0	8	12													
B08	705	45	350	90	robot&human	Screwdriver B0	Screwdriver B0	8	12													
B09	1055	45	350	90	robot&human	Screwdriver B0	Screwdriver B0	8	12													
B010	1415	250	370	500	robot&human	Screwdriver B0	Screwdriver B0	8	12													
CS0	800	500	1600	1000	human			40		B01	B02	B03	B04	B05	B06	B07	B08	B09	B010			
SL1	500	955	1000	90	human			6		CS0												
SL2	45	500	90	1000	human			6		CS0												
SL3	955	500	90	1000	human			6		CS0												
SL4	400	45	800	90	human			6		CS0												
SL5	1200	45	800	90	human			6		CS0												
PA1	1400	500	100	100	human			20		CS0												
CL1	1415	250	220	300	robot&human		Stick	10	12	PA1												
BM	1415	250	220	300	robot&human		Sucker	8	12	CL1												
MS1	500	955	1500	90	human			15		CS0												
MS2	500	45	1500	90	human			15		CS0												
MBS	1400	200	200	100	human			15		CS0												
CL2	375	955	750	90	human			5		CS0												
CL3	1125	45	750	90	human			5		CS0												
CL4	375	45	750	90	human			5		CS0												
CL5	1125	955	750	90	human			5		CS0												
CL6	1400	200	200	100	human			5		CS0												
HV1	500	45	1000	90	human			18		MS3	MS2	MS1	CL2	CL3	CL4	CL5	CL6					
HV2	500	955	1000	90	human			18		MS3	MS3	MS2	CL2	CL3	CL4	CL5	CL6					
CO1	1300	600	70	150	robot&human		Cover Gripper	5	10	CS0												
NO	1300	600	70	150	robot&human	Unscrewer N	Unscrewer N	10	13	CO1												
HVD	1400	600	150	200	robot&human		Gripper	10	13	BM	NO											
S1	1450	500	30	30	robot&human	Unscrewer S	Unscrewer S	12	16	CS0												
AS	1450	500	30	30	human			4		CS0												
TS1	375	955	750	90	human			18		CS0												
TS2	1125	45	750	90	human			18		CS0												
TS3	375	45	750	90	human			18		CS0												
TS4	1125	955	750	90	human			18		CS0												
TSS	1400	200	200	100	human			18		CS0												
LV1	500	45	1000	90	human			20		HVD	S1	TS1	TS2	TS3	TSA	TSS	AS					

Dati di Input sequenza di smontaggio ottimale

Figura 2: Dati di input nel file Excel (parte 2)

Codice	x0	y0	Length_x	Length_y	Operatore	Tool human	Tool robot	t ciclo human	t ciclo robot	REQUIREMENTS													
BR15	1550	250	100	300	robot&human		Sucker	12	18	CS2	N28												
N29	1200	750	750	750	robot&human	Unscrew N	Unscrew N	18	15	CS0													
BR16	1550	750	100	300	robot&human		Sucker	12	18	CS2	N29												
PA2	1415	80	340	90	human			18				BR11											
RC1	1415	80	340	90	human			18				BR11											
HVM1	1415	80	340	90	human			15				PA2	RC1										
PA3	1415	920	340	90	human			18				BR12											
RC2	1415	920	340	90	human			18				BR12											
HVM2	1415	920	340	90	human			15				PA3	RC2										
B17	1450	90	200	100	robot&human	Unscrew B	Unscrew B	17	14			BR11	BR12										
N66	1450	910	200	100	robot&human	Unscrew N	Unscrew N	6	4			CS0											
B54	1415	500	370	1000	human	Winch		220				BR15	BR16	HVM1	HVM2	B12	N66	HV1	HV2	LV1	LV2		
HT1	355	500	350	1000	robot&human		Sucker	7	12			BS1											
HT2	700	500	350	1000	robot&human		Sucker	7	12			BS2											
HT3	1055	500	350	1000	robot&human		Sucker	7	12			BS3											
B13	1415	500	370	1000	robot&human	Unscrew B	Unscrew B	13	9			BS3											
HT4	1415	500	370	1000	robot&human		Sucker	10	8			BS4	B13										
B14	160	500	100	500	robot&human	Unscrew B	Unscrew B	20	17			BS1	BS2										
BC1	160	500	100	1000	human	Winch		10				B14											
B15	470	500	100	500	robot&human	Unscrew B	Unscrew B	26	20			BS2											
BC2	470	500	100	1000	human	Winch		10				B15											
B16	850	500	100	500	robot&human	Unscrew B	Unscrew B	26	20			BS2											
BC3	850	500	100	1000	human	Winch		10				B16											
BC4	1200	500	100	1000	human	Winch		10				BS4	B13										
B17	700	250	355	500	robot&human	Unscrew B	Unscrew B	10	14			HT1											
B18	1055	250	355	500	robot&human	Unscrew B	Unscrew B	15	20			HT2											
B19	1415	250	355	500	robot&human	Unscrew B	Unscrew B	15	20			HT3											
B20	1415	500	355	1000	robot&human	Unscrew B	Unscrew B	10	14			HT4											
B21	700	750	355	500	robot&human	Unscrew B	Unscrew B	7	10			BC2											
B22	1055	750	355	500	robot&human	Unscrew B	Unscrew B	10	14			BC3											
B23	1415	750	355	500	robot&human	Unscrew B	Unscrew B	7	10			BC4											
CP2	800	500	1600	1000	human			30				BC1	B17	B18	B19	B20	B21	B22	B23				
IP1	355	500	350	1000	robot&human		Sucker	13	18			CP2											
IP2	700	500	350	1000	robot&human		Sucker	13	18			CP2											
IP3	1055	500	350	1000	robot&human		Sucker	13	18			CP2											
IP4	1415	500	370	1000	robot&human		Sucker	13	18			CP2											

Figura 3: Dati di input nel file Excel (parte 3)

Codice	x0	y0	Length_x	Length_y	Operatore	Tool human	Tool robot	t ciclo human	t ciclo robot	REQUIREMENTS													
CN1	30	800	50	200	human			8		CS0													
HN1	30	800	50	200	human	Spanner	Unscrew N2	8		CS0													
OH	90	500	160	500	human			12		CN1	HN1												
CN2	30	800	50	200	human			15		CS0													
HN2	30	800	50	200	human	Spanner	Unscrew N2	15		CS0													
N22	30	800	50	200	robot&human	Unscrew N	Unscrew N	6	9			RC											
IH	90	770	160	250	human			16		CN2	HN2												
HC1	1550	200	80	80	human	Spanner		30		CS0													
H1	1550	200	80	80	human			25		HC1													
HC2	1550	500	80	80	human	Spanner		30		CS0													
H2	1550	500	80	80	human			25		HC2													
HC3	1550	800	80	80	human	Spanner		30		CS0													
H3	1550	800	80	80	human			25		HC3													
HT5	1450	500	400	900	robot&human		Sucker	5	9			BS5											
N24	1450	500	400	900	robot&human	Unscrew N	Unscrew N	13	18			HT5											
CP1	1450	500	400	900	human			45		N24													
IP5	1450	500	400	900	robot&human		Sucker	6	5			CP1											
N25	1450	90	300	130	robot&human	Unscrew N	Unscrew N	15	20			CS0											
N56	1450	910	300	130	robot&human	Unscrew N	Unscrew N	15	20			CS0											
CS2	1450	500	400	900	human			45		IP5	N25	N56											
N26	1420	100	360	80	robot&human	Unscrew N	Unscrew N	23	20			CS0											
CH1	1420	100	360	80	human			9		CS0													
BR11	1420	100	360	80	robot&human		Sucker	12	17			N26	CH1	CS2									
N27	1420	900	360	80	robot&human	Unscrew N	Unscrew N	15	20			CS0											
CH2	1420	900	360	80	human			9		CS0													
BR12	1420	900	360	80	robot&human		Sucker	12	17			N27	CH2	CS2									
CO8	1055	450	350	50	robot&human		Cover Gripper	6	9			CS0											
N7	1055	450	350	50	robot&human	Unscrew N	Unscrew N	7	10			CO8											
BU7	1055	450	350	50	robot&human		Gripper	5	7			N7											
CO9	1055	900	350	50	robot&human		Cover Gripper	6	9			CS0											
N8	1055	900	350	50	robot&human	Unscrew N	Unscrew N	7	10			CO9											
B08	1055	900	350	50	robot&human		Gripper	6	10			N8											
B10	1200	250	100	300	robot&human	Unscrew B	Unscrew B	13	17			CS0											
BR13	1200	250	100	300	robot&human		Sucker	12	17			CS2	B10										
B11	1200	750	100	300	robot&human	Unscrew B	Unscrew B	13	17			CS0											
BR14	1200	750	100	300	robot&human		Sucker	12	17			CS2	B11										
N33	1055	80	340	90	robot&human	Unscrew N	Unscrew N	15	20			CS0											
N34	1055	80	340	90	robot&human	Unscrew N	Unscrew N	15	20			CS0											
BS3	1055	500	350	1000	human	Winch		220		BU7	B08	BR13	BR14	N33	HV1	HV2	LV1	LV2	N34				
N28	1200	250	250	250	robot&human	Unscrew N	Unscrew N	18	15			CS0											

Dati di Input sequenza di smontaggio ottimale

Figura 4: Dati di input nel file Excel (parte 4)

Codice	x0	y0	Length_x	Length_y	Operatore	Tool human	Tool robot	t ciclo human	t ciclo robot	REQUIREMENTS														
CO3	150	550	100	150	robot&human		Cover Gripper	5	8	CS0														
N2	150	550	100	150	robot&human	Unscrew N	Unscrew N	4	6	CO3														
BU2	150	550	100	150	robot&human		Gripper	5	10	S4	N2													
CO4	355	470	350	50	robot&human		Cover Gripper	6	9	CS0														
N3	355	470	350	50	robot&human	Unscrew N	Unscrew N	7	10	CO4														
BU3	355	470	350	50	robot&human		Gripper	5	7	N3														
N16	90	450	180	350	robot&human	Unscrew N	Unscrew N	12	18	CS0														
N17	90	450	180	350	robot&human	Unscrew N	Unscrew N	4	6	BU3														
N18	90	450	180	350	robot&human	Unscrew N	Unscrew N	4	6	BU1														
RC	90	450	180	350	robot&human		Gripper	6	8	N16	N17	N18	BU2	AC	AS	TN								
CO5	355	900	350	50	robot&human		Cover Gripper	6	9	CS0														
N4	355	900	350	50	robot&human	Unscrew N	Unscrew N	7	10	CO5														
BU4	355	900	350	50	robot&human		Gripper	5	7	N4														
B3	470	250	100	300	robot&human	Unscrew B	Unscrew B	10	7	CS0														
BR3	470	250	100	300	robot&human		Sucker	8	12	B3														
B4	470	250	100	300	robot&human	Unscrew B	Unscrew B	10	7	CS0														
BR4	470	250	100	300	robot&human		Sucker	8	12	B4														
B7	160	250	100	300	robot&human	Unscrew B	Unscrew B	13	10	CS0														
BR7	160	250	100	300	robot&human		Sucker	12	10	B7														
BR8	160	250	100	300	robot&human	Unscrew B	Unscrew B	13	10	RC														
BR8	160	250	100	300	robot&human		Sucker	13	10	BR8														
N30	330	80	340	90	robot&human	Unscrew N	Unscrew N	12	15	CS0														
N42	330	920	340	90	robot&human	Unscrew N	Unscrew N	12	18	CS0														
N31	160	240	50	50	robot&human	Unscrew N	Unscrew N	5	7	RC														
BS1	355	500	350	1000	human	Winch		220		BU1	BU3	BU4	BR3	BR4	BR7	BR8	N30	N31	N42	HV1	HV2	LV1	LV2	
CO6	700	140	350	50	robot&human		Cover Gripper	6	9	CS0														
N5	700	140	350	50	robot&human	Unscrew N	Unscrew N	7	10	CO6														
BU5	700	140	350	50	robot&human		Gripper	7	14	N5														
CO7	700	640	350	50	robot&human		Cover Gripper	6	9	CS0														
N6	700	640	350	50	robot&human	Unscrew N	Unscrew N	7	10	CO7														
BU6	700	640	350	50	robot&human		Gripper	7	14	N6														
B5	850	250	100	300	robot&human	Unscrew B	Unscrew B	14	20	CS0														
BR5	850	250	100	300	robot&human		Sucker	7	14	B5														
B6	850	250	100	300	robot&human	Unscrew B	Unscrew B	14	20	CS0														
BR6	850	250	100	300	robot&human		Sucker	7	14	B6														
N32	700	80	340	90	robot&human	Unscrew N	Unscrew N	14	20	CS0														
N50	700	80	340	90	robot&human	Unscrew N	Unscrew N	14	20	CS0														
BS2	700	500	350	1000	human	Winch		220		BU5	BU6	BR3	BR4	BR5	BR6	N32	N50	HV1	HV2	LV1	LV2			
B9	30	120	50	200	robot&human	Unscrew B	Unscrew B	13	22	RC														
EC2	30	120	50	200	human	Unscrew B		8		B9														

Figura 5: Dati di input nel file Excel (parte 5)

Codice	x0	y0	Length_x	Length_y	Operatore	Tool human	Tool robot	t ciclo human	t ciclo robot	REQUIREMENTS														
LV2	500	955	1000	90	human			20		HVD	S1	TS1	TS2	TS3	TS4	TS5	AS							
N11	1550	250	100	250	robot&human	Unscrew N	Unscrew N	18	15	CS0														
N12	1550	250	100	250	human	Unscrew N	Unscrew N	18		CS0														
BR1	1550	250	100	250	robot&human		Sucker	8	12	N11	N12													
N13	1550	700	100	250	robot&human	Unscrew N	Unscrew N	18	15	CS0														
N14	1550	700	100	250	human	Unscrew N	Unscrew N	18		CS0														
BR2	1550	700	100	250	robot&human		Sucker	8	12	N13	N14													
N19	1450	90	400	130	robot&human	Unscrew N	Unscrew N	15	18	CS0														
N40	1450	910	400	130	robot&human	Unscrew N	Unscrew N	15	18	CS0														
CS1	1450	500	400	900	human			30		N19	LV1	LV2	HV1	HV2	BR1	BR2	N40							
CO10	1150	200	100	100	robot&human		Cover Gripper	5	9	CS0														
N9	1150	200	100	100	robot&human	Unscrew N	Unscrew N	6	10	CO10														
BU9	1150	200	100	100	robot&human		Gripper	7	11	N9														
CO11	1150	750	100	400	robot&human		Cover Gripper	5	9	CS0														
N10	1150	750	100	400	robot&human	Unscrew N	Unscrew N	6	10	CO11														
BU10	1150	750	100	400	robot&human		Gripper	7	11	N10														
N20	1200	200	120	250	robot&human	Unscrew N	Unscrew N	31	40	CS0														
BR9	1200	200	120	250	robot&human		Sucker	8	12	CS1	BU9	BU10	N20											
N21	1200	750	120	250	robot&human	Unscrew N	Unscrew N	31	40	CS0														
BR10	1200	750	120	250	robot&human		Sucker	8	12	CS1	BU9	BU10	N21											
N23	1450	90	300	130	robot&human	Unscrew N	Unscrew N	10	15	CS0														
N45	1450	910	300	130	robot&human	Unscrew N	Unscrew N	10	15	CS0														
BS5	1415	500	370	1000	human	Winch		220		BR9	BR10	N23	N45											
S2	60	450	120	350	robot&human	Unscrew S	Unscrew S	12	16	CS0														
RE	90	450	180	350	robot&human		Sucker	8	5	S2														
B2	30	120	50	200	human	Unscrew B	Unscrew B	7		CS0														
EC1	30	120	50	200	human			6		B2														
B1	30	800	50	200	robot&human	Unscrew B	Unscrew B	4	10	CS0														
N15	30	800	50	200	robot&human	Unscrew N	Unscrew N	8	12	RE														
AC	30	800	50	200	human			8		B1	N15													
S5	30	120	50	200	robot&human	Unscrew S	Unscrew S	3	6	RE														
TP	30	120	50	200	human	Unscrew S	Unscrew S	5		S5	EC1													
S6	30	120	50	200	robot&human	Unscrew S	Unscrew S	3	6	RE														
TN	30	120	50	200	human			5		S6	TP													
S3	150	150	100	300	robot&human	Unscrew S	Unscrew S	8	8	CS0														
CO2	150	150	100	300	robot&human		Cover Gripper	5	8	CS0														
N1	150	150	100	300	robot&human	Unscrew N	Unscrew N	4	6	CO2														
BU1	150	150	100	300	robot&human		Gripper	5	10	S3	N1													
S4	150	550	100	150	robot&human	Unscrew S	Unscrew S	8	8	CS0														
CO3	150	550	100	150	robot&human		Cover Gripper	5	8	CS0														

Sequenza di smontaggio Audi Q5

Figura 6: Sequenza di smontaggio della batteria di un Audi Q5

Disassembly step number	Disassembly step	Necessary tool
I	Unscrew covers (1), (6) and casing bottom (12)	Screw driver
II	Removal of the power electronics cover (1)	Hand
III	Checking of the state of charge	Special measuring device
IV	Removal of the side covering (2)	Hand
V	Disassembly of the live lines from the modules/stacks (14)	Screw driver
VI	Cutting of the cable ties (3)	Side cutters
VII	Disassembly of the plug connection between the cell controllers and the BMS (4)	Hand
VIII	Removal of the BMS (4)	Hand
IX	Removal of the power electronics (5)	Hand
X	Cutting of the bus for the thermo sensors	Side cutters
XI	Disassembly of the system cover (6)	Screw driver
XII	Removal of the system cover (6)	Hand
XIII	Unscrew of the cable guiding (7)	Screw driver
XIV	Removal of the cable guiding (7)	Hand
XV	Removal of the gas venting (8)	Hand
XVI	Removal of the cover of stacks (9)	Hand
XVII	Disassembly of the connectors between the stacks (10)	Screw driver
XVIII	Removal of the connectors between the stacks (10)	Hand
XIX	Unscrew of the stack holders (11)	Screw driver
XX	Removal of the stack holders (11)	Hand
XXI	Removal of the casing bottom (12)	Hand
XXII	Unscrew of the stack fastener (13)	Screw driver
XXIII	Removal of the stack fastener (13)	Hand
XXIV	Removal of stacks (14)	Hand